Lecture Notes in Computer Science 4707

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
University of Dortmund, Germany

Madhu Sudan
Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos
University of California, Los Angeles, CA, USA

Doug Tygar
University of California, Berkeley, CA, USA

Moshe Y. Vardi
Rice University, Houston, TX, USA

Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Osvaldo Gervasi Marina L. Gavrilova (Eds.)

Computational Science and Its Applications – ICCSA 2007

International Conference
Kuala Lumpur, Malaysia, August 26-29, 2007
Proceedings, Part III

Springer

Volume Editors

Osvaldo Gervasi
University of Perugia, Department of Mathematics and Computer Science
Via Vanvitelli, 1, 06123 Perugia, Italy
E-mail: osvaldo@unipg.it

Marina L. Gavrilova
University of Calgary, Department of Computer Science
500 University Dr. N.W., Calgary, AB, Canada
E-mail: marina@cpsc.ucalgary.ca

Associated Editors:

David Taniar
Monash University, Clayton, Australia

Andrès Iglesias
University of Cantabria, Santander, Spain

Antonio Laganà
University of Perugia, Italy

Deok-Soo Kim
Hanyang University, Seoul, Korea

Youngsong Mun
Soongsil University, Seoul, Korea

Hyunseung Choo
Sungkyunkwan University, Suwon, Korea

Library of Congress Control Number: 2007933006

CR Subject Classification (1998): F, D, G, H, I, J, C.2-3

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN	0302-9743
ISBN-10	3-540-74482-7 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-74482-5 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2007
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12112180 06/3180 5 4 3 2 1 0

Preface

This three volume set constitutes the proceedings of the 2007 International Conference on Computational Science and its Applications, ICCSA 2007, held in Kuala Lumpur, Malaysia, from August 26–29, 2007. It represents a comprehensive collection of 300 refereed full papers selected from approximately 1,250 submissions to ICCSA 2007.

The continuous support of computational science researchers has helped ICCSA to become a firmly established forum in the area of scientific computing. This year, the collection of fully refereed high-quality original works accepted as long papers for presentation at ICCSA 2007 have been published in this LNCS volume. This outstanding collection complements the volume of short papers, published for the first time by IEEE CS. All of the long papers presented in this collection of volumes share a common theme: computational science.

Over the past ten years, since the first conference on computational science took place, this vibrant and promising area has firmly established itself as a vital part of many scientific investigations in a broad gamut of disciplines. Having deep roots in fundamental disciplines, such as mathematics, physics, and chemistry, the computational science field is finding new applications in such broad and diverse areas as aerospace and automotive industries, bioinformatics and nanotechnology studies, networks and grid computing, computational geometry and biometrics, computer education, and art. Due to the growing complexity and sophistication of many challenges in computational science, the use of sophisticated algorithms and emerging technologies is inevitable. Together, these far reaching scientific areas help to shape this conference in the realms of state-of-the-art computational science research and applications, encompassing the facilitating theoretical foundations and the innovative applications of such results in other areas.

The topics of the short refereed papers presented in this volume span all the traditional as well as the emerging computational science areas, and are structured according to the major conference themes:

- Computational Methods, Algorithms and Applications
- High Performance Technical Computing and Networks
- Advanced and Emerging Applications
- Geometric Modeling, Graphics and Visualization
- Information Systems and Information Technologies

Moreover, selected short papers from 30 workshops and technical sessions on such areas as information security, web learning, software engineering, computational intelligence, digital security, mobile communications, grid computing, modeling, optimization, embedded systems, wireless networks, computational geometry, computer graphics, biometrics, molecular structures, geographical information systems, ubiquitous computing, symbolic computations, molecular

structures, web systems and intelligence, e-printing, and education are included in this publication.

We are very grateful to the International Steering Committee and the International Program Committee for their tremendous support in putting this conference together, the nearly four hundred referees for their diligent work in reviewing the submissions, and all the sponsors, supporting organizations and volunteers of ICCSA for contributing their time, energy and resources to this event.

Finally, we thank all authors for their submissions making the ICCSA conference year after year one of the premium events on the scientific community scene, facilitating the exchange of ideas, fostering new collaborations, and shaping the future of computational science.

August 2007

Osvaldo Gervasi
Marina L. Gavrilova

Organization

ICCSA 2007 was organized by the University of Perugia (Italy), the University of Calgary (Canada) and the Universiti Teknologi Malaysia (Malaysia).

Conference Chairs

Marina L. Gavrilova (University of Calgary, Calgary, Canada), Scientific Chair
Osvaldo Gervasi (University of Perugia, Perugia, Italy), Program Chair

Steering Committee

Alexander V. Bogdanov (Institute for High Performance Computing and Data Bases, Russia)
Hyunseung Choo (Sungkyunkwan University, Korea)
Marina L. Gavrilova (University of Calgary, Canada)
Osvaldo Gervasi (University of Perugia, Perugia, Italy)
Andres Iglesias (University of Cantabria, Spain)
Vipin Kumar (Army High Performance Computing Center and University of Minnesota, USA)
Antonio Laganà (University of Perugia, Italy)
Youngsong Mun (Soongsil University, Korea)
C.J. Kenneth Tan (OptimaNumerics, UK)
David Taniar (Monash University, Australia)

Session Organizers

Advanced Security Services (ASS 07)

Eui-Nam Huh, Kyung Hee University (Korea)

Advances in Web Based Learning (AWBL 07)

Mustafa Murat Inceoglu and Eralp Altun, Ege University (Turkey)

CAD/CAM and Web Based Collaboration (CADCAM 07)

Yongju Cho, KITECH (Korea)
Changho Lee, Yonsei University (Korea)

Component Based Software Engineering and Software Process Models (CBSE 07)

Haeng-Kon Kim, Daegu University (Korea)

Computational Geometry and Applications (CGA 07)

Marina Gavrilova, University of Calgary (Canada)

Computational Intelligence Approaches and Methods for Security Engineering (CIAMSE 07)

Tai-hoon Kim, Ewha Womans University and SERC (Korea)
Haeng-kon Kim, Catholic University of Daegu (Korea)

Computational Linguistics (CL 07)

Hyungsuk Ji, Sungkyunkwan University (Korea)

Digital Content Security and Management of Distributed Computing (DCSMDC 07)

Geuk Lee, Hannam University (Korea)

Distributed Data and Storage System Management (DDSM 07)

Jemal Abawajy, Deakin University (Australia)
Maria Pérez, Universidad Politécnica de Madrid (Spain)
Laurence T. Yang, St. Francis Xavier University (Canada)

Data Storage Device and Systems (DS2 07)

Yeonseung Ryu, Myongji University (Korea)

e-Printing CAE Technology (E-PCAET 07)

Seoung Soo Lee, Konkuk University (Korea)

Embedded Systems for Ubiquitous Computing (ESUC 07)

Jiman Hong, Kwangwoon University (Korea)
Tei-Wei Kuo, National Taiwan University (Taiwan)

High-Performance Computing and Information Visualization (HPCIV 07)

Frank Devai, London South Bank University (UK)
David Protheroe, London South Bank University (UK)

Integrated Analysis and Intelligent Design Technology (IAIDT 07)

Jae-Woo Lee, CAESIT and Konkuk University (Korea)

Intelligent Image Mining (IIM 07)

Hyung-Il Choi, Soongsil University (Korea)

Intelligence and Security Informatics (ISI 07)

Kuinam J. Kim and Donghwi Lee, Kyonggi University (Korea)

Information Systems and Information Technologies (ISIT 07)

Youngsong Mun, Soongsil University (Korea)

Mobile Communications (MobiComm 07)

Hyunseung Choo, Sungkyunkwan University (Korea)

Molecular Simulations Structures and Processes (MOSSAP 07)

Antonio Laganà, University of Perugia (Italy)

Middleware Support for Distributed Computing (MSDC 07)

Sung Y. Shin, South Dakota State University (USA)
Jaeyoung Choi, Soongsil University (Korea)

Optimization: Theory and Applications (OTA 07)

Dong-Ho Lee, Hanyang University (Korea)
Ertugrul Karsak, Galatasaray University (Turkey)
Deok-Soo Kim, Hanyang University (Korea)

Pattern Recognition and Ubiquitous Computing (PRUC 07)

Jinok Kim, Daegu Haany University (Korea)

PULSES - Logical, Technical and Computational Aspects of Transformations and Suddenly Emerging Phenomena (PULSES 07)

Carlo Cattani, University of Salerno (Italy)
Cristian Toma, University of Bucarest (Romania)

Technical Session on Computer Graphics (TSCG 07)

Andres Iglesias, University of Cantabria Santander (Spain)
Deok-Soo Kim, Hanyang University, Seoul (Korea)

Ubiquitous Applications & Security Service (UASS 07)

Hai Jin, Huazhong University of Science and Technology (China)
Yeong-Deok Kim, Woosong University (Korea)

Virtual Reality in Scientific Applications and Learning (VRSAL 07)

Osvaldo Gervasi, University of Perugia (Italy)

Wireless and Ad-Hoc Networking (WAD 07)

Jongchan Lee and Sangjoon Park, Kunsan National University (Korea)

Workshop on Internet Communication Security (WICS 07)

José Maria Sierra Camara, University of Madrid (Spain)

Wireless Sensor Networks (WSNs 07)

Jemal Abawajy, Deakin University (Australia)
David Taniar, Monash University (Australia)
Mustafa Mat Deris, University College of Science and Technology (Malaysia)
Laurence T. Yang, St. Francis Xavier University (Canada)

Program Committee

Jemal Abawajy (Deakin University, Australia)
Kenny Adamson (EZ-DSP, UK)
Frank Baetke (Hewlett Packard, USA)
Mark Baker (Portsmouth University, UK)
Young-Cheol Bang (Korea Politechnic University, Korea)
David Bell (The Queen's University of Belfast, UK)
J.A. Rod Blais (University of Calgary, Canada)
Alexander V. Bogdanov (Institute for High Performance Computing and Data Bases, Russia)
John Brooke (University of Manchester, UK)
Martin Buecker (Aachen University, Germany)
Yves Caniou (INRIA, France)
YoungSik Choi (University of Missouri, USA)
Hyunseung Choo (Sungkyunkwan University, Korea)
Min Young Chung (Sungkyunkwan University, Korea)
Yiannis Cotronis (University of Athens, Greece)
Jose C. Cunha (New University of Lisbon, Portugal)
Alexander Degtyarev (Institute for High Performance Computing and Data Bases, Russia)
Tom Dhaene (University of Antwerp, Belgium)
Beniamino Di Martino (Second University of Naples, Italy)
Hassan Diab (American University of Beirut, Lebanon)
Marina L. Gavrilova (University of Calgary, Canada)
Michael Gerndt (Technical University of Munich, Germany)
Osvaldo Gervasi (University of Perugia, Italy)
Christopher Gold (Hong Kong Polytechnic University, Hong Kong)
Yuriy Gorbachev (Institute of High Performance Computing and Information Systems, Russia)
Andrzej Goscinski (Deakin University, Australia)
Ladislav Hluchy (Slovak Academy of Science, Slovakia)
Eui-Nam John Huh (Seoul Woman's University, Korea)
Shen Hong (Japan Advanced Institute of Science and Technology, Japan)
Terence Hung (Institute of High Performance Computing, Singapore)
Andres Iglesias (University of Cantabria, Spain)
Peter K Jimack (University of Leeds, UK)
Benjoe A. Juliano (California State University at Chico, USA)
Peter Kacsuk (MTA SZTAKI Research Institute, Hungary)
Kyung Wo Kang (KAIST, Korea)
Daniel Kidger (Quadrics, UK)
Haeng Kon Kim (Catholic University of Daegu, Korea)
Jin Suk Kim (KAIST, Korea)
Tai-Hoon Kim (Korea Information Security Agency, Korea)

Yoonhee Kim (Syracuse University, USA)
Dieter Kranzlmueller (Johannes Kepler University Linz, Austria)
Deok-Soo Kim (Hanyang University, Korea)
Antonio Laganà (University of Perugia, Italy)
Francis Lau (The University of Hong Kong, Hong Kong)
Bong Hwan Lee (Texas A&M University, USA)
Dong Chun Lee (Howon University, Korea)
Sang Yoon Lee (Georgia Institute of Technology, USA)
Tae-Jin Lee (Sungkyunkwan University, Korea)
Yong Woo Lee (University of Edinburgh, UK)
Bogdan Lesyng (ICM Warszawa, Poland)
Er Ping Li (Institute of High Performance Computing, Singapore)
Laurence Liew (Scalable Systems Pte, Singapore)
Chun Lu (Institute of High Performance Computing, Singapore)
Emilio Luque (Universitat Autònoma de Barcelona, Spain)
Michael Mascagni (Florida State University, USA)
Graham Megson (University of Reading, UK)
John G. Michopoulos (US Naval Research Laboratory, USA)
Byoung Joon Min (U.C. Irvine, USA)
Edward Moreno (Euripides Foundation of Marilia, Brazil)
Youngsong Mun (Soongsil University, Korea)
Jiri Nedoma (Academy of Sciences of the Czech Republic, Czech Republic)
Salvatore Orlando (University of Venice, Italy)
Robert Panoff (Shodor Education Foundation, USA)
Marcin Paprzycki (Oklahoma State University, USA)
Gyung-Leen Park (University of Texas, USA)
Ron Perrott (The Queen's University of Belfast, UK)
Dimitri Plemenos (University of Limoges, France)
Richard Ramaroson (ONERA, France)
Rosemary Renaut (Arizona State University, USA)
Alistair Rendell (Australian National University, Australia)
Alexey S. Rodionov (Russian Academy of Sciences, Russia)
Paul Roe (Queensland University of Technology, Australia)
Heather J. Ruskin (Dublin City University, Ireland)
Muhammad Sarfraz (King Fahd University of Petroleum and Minerals, Saudi Arabia)
Siti Mariyam Shamsuddin (Universiti Technologi Malaysia, Malaysia)
Jie Shen (University of Michigan, USA)
Dale Shires (US Army Research Laboratory, USA)
Jose Sierra-Camara (University Carlos III of Madrid, Spain)
Vaclav Skala (University of West Bohemia, Czech Republic)
Alexei Sourin (Nanyang Technological University, Singapore)
Olga Sourina (Nanyang Technological University, Singapore)
Elena Stankova (Institute for High Performance Computing and Data Bases, Russia)

Gunther Stuer (University of Antwerp, Belgium)
Kokichi Sugihara (University of Tokyo, Japan)
Boleslaw Szymanski (Rensselaer Polytechnic Institute, USA)
Ryszard Tadeusiewicz (AGH University of Science and Technology, Poland)
C. J. Kenneth Tan (OptimaNumerics, UK, and The Queen's University of Belfast, UK)
David Taniar (Monash University, Australia)
Ruppa K. Thulasiram (University of Manitoba, Canada)
Pavel Tvrdik (Czech Technical University, Czech Republic)
Putchong Uthayopas (Kasetsart University, Thailand)
Mario Valle (Swiss National Supercomputing Centre, Switzerland)
Marco Vanneschi (University of Pisa, Italy)
Piero Giorgio Verdini (University of Pisa and Istituto Nazionale di Fisica Nucleare, Italy)
Jesus Vigo-Aguiar (University of Salamanca, Spain)
Jens Volkert (University of Linz, Austria)
Koichi Wada (University of Tsukuba, Japan)
Ping Wu (Institute of High Performance Computing, Singapore)
Jinchao Xu (Pennsylvania State University, USA)
Chee Yap (New York University, USA)
Osman Yasar (SUNY at Brockport, USA)
George Yee (National Research Council and Carleton University, Canada)
Yong Xue (Chinese Academy of Sciences, China)
Myung Sik Yoo (SUNY, USA)
Igor Zacharov (SGI Europe, Switzerland)
Alexander Zhmakin (SoftImpact, Russia)
Zahari Zlatev (National Environmental Research Institute, Denmark)
Albert Zomaya (University of Sydney, Australia)

Local Organizing Committee

Alias Abdul-Rahman (Universiti Teknologi Malaysia, Chair)
Mohamad Nor Said (Universiti Teknologi Malaysia)
Zamri Ismail (Universiti Teknologi Malaysia)
Zulkepli Majid (Universiti Teknologi Malaysia)
Muhammad Imzan Hassan (Universiti Teknologi Malaysia)
Ivin Amri Musliman (Universiti Teknologi Malaysia)
Chen Tet Khuan (Universiti Teknologi Malaysia)
Harith Fadzilah Khalid (Universiti Teknologi Malaysia)
Mohd Hasif Nasruddin (Universiti Teknologi Malaysia)
Mohd Hafiz Sharkawi (Universiti Teknologi Malaysia)
Muhamad Uznir Ujang (Universiti Teknologi Malaysia)
Siti Awanis Zulkefli (Universiti Teknologi Malaysia)

Venue

ICCSA 2007 took place in the magnificent Sunway Hotel and Resort in Kuala Lumpur, Malaysia

Sunway Hotel & Resort
Persiaran Lagoon, Bandar Sunway
Petaling Jaya 46150
Selangor Darul Ehsan
Malaysia

Sponsoring Organizations

ICCSA 2007 would not have been possible without the tremendous support of many organizations and institutions, for which all organizers and participants of ICCSA 2007 express their sincere gratitude:

University of Perugia, Italy
University of Calgary, Canada
OptimaNumerics, UK
Spark Planner Pte Ltd, Singapore
SPARCS Laboratory, University of Calgary, Canada
MASTER-UP, Italy

Table of Contents – Part III

Workshop on CAD/CAM and Web Based Collaboration (CADCAM 07)

Framework of Integrated System for the Innovation of Mold Manufacturing Through Process Integration and Collaboration 1
Bo Hyun Kim, Sung Bum Park, Gyu Bong Lee, and So Young Chung

A Study on Automated Design System for a Blow Mould 11
Yong Ju Cho, Kwang Yeol Ryu, and Seok Woo Lee

Development of an Evaluation System of the Informatization Level for the Mould Companies in Korea 20
Yong Ju Cho and Sung Hee Lee

Framework of a Collaboration-Based Engineering Service System for Mould Industry 33
Chang Ho Lee and Yong Ju Cho

Workshop on Component Based Software Engineering and Software Process Model (CBSE 07)

Meta-modelling Syntax and Semantics of Structural Concepts for Open Networked Enterprises 45
Mohamed Bouhdadi, Youssef Balouki, and El maati Chabbar

Component Specification for Parallel Coupling Infrastructure 55
J. Walter Larson and Boyana Norris

Real-Time Navigation for a Mobile Robot Based on the Autonomous Behavior Agent 69
Lu Xu, Liguo Zhang, and Yangzhou Chen

Concurrent Subsystem-Component Development Model (CSCDM) for Developing Adaptive E-Commerce Systems 81
Liangtie Dai and Wanwu Guo

A Quantitative Approach for Ranking Change Risk of Component-Based Software 92
Chengying Mao

Relating Software Architecture Views by Using MDA 104
Rogelio Limon Cordero and Isidro Ramos Salavert

Workshop on Distributed Data and Storage System Managemnt (DDSM 07)

Update Propagation Technique for Data Grid 115
Mohammed Radi, Ali Mamat, M. Mat Deris, Hamidah Ibrahim, and Subramaniam Shamala

A Spatiotemporal Database Prototype for Managing Volumetric Surface Movement Data in Virtual GIS 128
Mohd Shafry Mohd Rahim, Abdul Rashid Mohamed Shariff, Shattri Mansor, Ahmad Rodzi Mahmud, and Daut Daman

Query Distributed Ontology over Grid Environment 140
Ngot Phu Bui, SeungGwan Lee, and TaeChoong Chung

Workshop on Embedded Systems for Ubiquitous Computing (ESUC 07)

CSP Transactors for Asynchronous Transaction Level Modeling and IP Reuse 154
Lilian Janin and Doug Edwards

A Robust Real-Time Message Scheduling Scheme Capable of Handling Channel Errors in Wireless Local Area Networks 169
Junghoon Lee, Mikyung Kang, Gyung-Leen Park, In-Hye Shin, Hanil Kim, and Sang-Wook Kim

Design and Implementation of a Tour Planning System for Telematics Users 179
Junghoon Lee, Euiyoung Kang, and Gyung-Leen Park

General Track

Ionospheric F-Layer Critical Frequency Estimation from Digital Ionogram Analysis 190
Nipon Theera-Umpon

Study of Digital License Search for Intellectual Property Rights of S/W Source Code 201
Byungrae Cha, Kyungjun Kim, and Dongseob Lee

Creating Numerically Efficient FDTD Simulations Using Generic C++ Programming 213
I. Valuev, A. Deinega, A. Knizhnik, and B. Potapkin

Mutual Authentication Protocol for RFID Tags Based on Synchronized Secret Information with Monitor 227
Song Han, Vidyasagar Potdar, and Elizabeth Chang

Non-linear Least Squares Features Transformation for Improving the Performance of Probabilistic Neural Networks in Classifying Human Brain Tumors on MRI 239
Pantelis Georgiadis, Dionisis Cavouras, Ioannis Kalatzis, Antonis Daskalakis, George Kagadis, Koralia Sifaki, Menelaos Malamas, George Nikiforidis, and Ekaterini Solomou

Adaptive Scheduling for Real-Time Network Traffic Using Agent-Based Simulation 248
Moutaz Saleh and Zulaiha Ali Othman

Defining Security Architectural Patterns Based on Viewpoints 262
David G. Rosado, Carlos Gutiérrez, Eduardo Fernández-Medina, and Mario Piattini

A New Nonrepudiable Threshold Proxy Signature Scheme with Valid Delegation Period 273
Min-Shiang Hwang, Shiang-Feng Tzeng, and Chun-Ta Li

Two-Stage Interval Krawczyk-Schwarz Methods with Applications to Nonlinear Parabolic PDE 285
Hartmut Schwandt

Red-Black EDG*SOR* Iterative Method Using Triangle Element Approximation for 2D Poisson Equations 298
J. Sulaiman, M. Othman, and M.K. Hasan

Performance of Particle Swarm Optimization in Scheduling Hybrid Flow-Shops with Multiprocessor Tasks 309
M. Fikret Ercan and Yu-Fai Fung

Branch-and-Bound Algorithm for Anycast Flow Assignment in Connection-Oriented Networks 319
Krzysztof Walkowiak

Quasi-hierarchical Evolutionary Algorithm for Flow Optimization in Survivable MPLS Networks 330
Michał Przewoźniczek and Krzysztof Walkowiak

An Exact Algorithm for the Minimal Cost Gateways Location, Capacity and Flow Assignment Problem in Two-Level Hierarchical Wide Area Networks 343
Przemyslaw Ryba and Andrzej Kasprzak

Implementing and Optimizing a Data-Intensive Hydrodynamics Application on the Stream Processor 353
Ying Zhang, Gen Li, and Xuejun Yang

On Disconnection Node Failure and Stochastic Static Resilience of P2P Communication Networks 367
F. Safaei, M. Fathy, A. Khonsari, and N. Talebanfard

An Efficient Sequence Alignment Algorithm on a LARPBS 379
David Semé and Sidney Youlou

An Effective Unconditionally Stable Algorithm for Dispersive Finite Difference Time Domain Simulations 388
Omar Ramadan

A Novel Congestion Control Scheme for Elastic Flows in Network-on-Chip Based on Sum-Rate Optimization 398
Mohammad S. Talebi, Fahimeh Jafari, Ahmad Khonsari, and Mohammad H. Yaghmae

3D Bathymetry Reconstruction from Airborne Topsar Polarized Data 410
Maged Marghany, Mazlan Hashim, and Arthur P. Cracknell

A Parallel FDTD Algorithm for the Solution of Maxwell's Equations with Nearly PML Absorbing Boundary Conditions 421
Omar Ramadan

Application of Modified ICA to Secure Communications in Chaotic Systems 431
Shih-Lin Lin and Pi-Cheng Tung

Zero Memory Information Sources Approximating to Video Watermarking Attacks 445
M. Mitrea, O. Dumitru, F. Prêteux, and A. Vlad

On Statistical Independence in the Logistic Map: A Guide to Design New Chaotic Sequences Useful in Cryptography 460
Adriana Vlad, Adrian Luca, and Bogdan Badea

FVM- and FEM-Solution of Elliptical Boundary Value Problems in Different Coordinate Systems 475
Günter Bärwolff

Digital Simulation for Micro Assembly Arranged at Rectangular Pattern in Micro Factory 486
Murali Subramaniyam, Sangho Park, Sung-il Choi, Seokho Jang, and Joon-Yub Song

A New Quantized Input RLS, QI-RLS, Algorithm 495
A. Amiri, M. Fathy, M. Amintoosi, and H. Sadoghi

Decentralized Replica Exchange Parallel Tempering: An Efficient Implementation of Parallel Tempering Using MPI and SPRNG 507
Yaohang Li, Michael Mascagni, and Andrey Gorin

Approximation Algorithms for 2-Source Minimum Routing Cost k-Tree Problems 520
Yen Hung Chen, Gwo-Liang Liao, and Chuan Yi Tang

On the Expected Value of a Number of Disconnected Pairs of Nodes in Unreliable Network 534
Alexey S. Rodionov, Olga K. Rodionova, and Hyunseung Choo

Linearization of Stream Ciphers by Means of Concatenated Automata 544
A. Fúster-Sabater and P. Caballero-Gil

Effective Quantification of Gene Expression Levels in Microarray Images Using a Spot-Adaptive Compound Clustering-Enhancement-Segmentation Scheme 555
Antonis Daskalakis, Dionisis Cavouras, Panagiotis Bougioukos, Spiros Kostopoulos, Pantelis Georgiadis, Ioannis Kalatzis, George Kagadis, and George Nikiforidis

Biomarker Selection, Employing an Iterative Peak Selection Method, and Prostate Spectra Characterization for Identifying Biomarkers Related to Prostate Cancer 566
Panagiotis Bougioukos, Dionisis Cavouras, Antonis Daskalakis, Ioannis Kalatzis, George Nikiforidis, and Anastasios Bezerianos

Classic Cryptanalysis Applied to Exons and Introns Prediction 575
Manuel Aguilar R., Héctor Fraire H., Laura Cruz R., Juan J. González B., Guadalupe Castilla V., and Claudia G. Gómez S.

Chronic Hepatitis and Cirrhosis Classification Using SNP Data, Decision Tree and Decision Rule 585
Dong-Hoi Kim, Saangyong Uhmn, Young-Woong Ko, Sung Won Cho, Jae Youn Cheong, and Jin Kim

Reconstruction of Suboptimal Paths in the Constrained Edit Distance Array with Application in Cryptanalysis 597
Slobodan Petrović and Amparo Fúster-Sabater

Solving a Practical Examination Timetabling Problem: A Case Study 611
Masri Ayob, Ariff Md Ab Malik, Salwani Abdullah, Abdul Razak Hamdan, Graham Kendall, and Rong Qu

A Geometric Design of Zone-Picking in a Distribution Warehouse 625
Ying-Chin Ho, Hui Ming Wee, and Hsiao Ching Chen

Routing Path Generation for Reliable Transmission in Sensor Networks Using GA with Fuzzy Logic Based Fitness Function 637
Jin Myoung Kim and Tae Ho Cho

A Heuristic Local Search Algorithm for Unsatisfiable Cores Extraction 649
Jianmin Zhang, Shengyu Shen, and Sikun Li

ontoX - A Method for Ontology-Driven Information Extraction 660
Burcu Yildiz and Silvia Miksch

Improving the Efficiency and Efficacy of the K-means Clustering Algorithm Through a New Convergence Condition 674
Joaquín Pérez O., Rodolfo Pazos R., Laura Cruz R., Gerardo Reyes S., Rosy Basave T., and Héctor Fraire H.

Modelling Agent Strategies in Simulated Market Using Iterated Prisoner's Dilemma 683
Raymond Chiong

A Local Search Algorithm for a SAT Representation of Scheduling Problems 697
Marco Antonio Cruz-Chávez and Rafael Rivera-López

A Context-Aware Solution for Personalized En-route Information Through a P2P Agent-Based Architecture 710
José Santa, Andrés Muñoz, and Antonio F.G. Skarmeta

A Survey of Revenue Models for Current Generation Social Software's Systems 724
Kevin Chai, Vidyasagar Potdar, and Elizabeth Chang

Context-Driven Requirements Analysis 739
Jongmyung Choi

Performance Analysis of Child/Descendant Queries in an XML-Enabled Database 749
Eric Pardede, J. Wenny Rahayu, David Taniar, and Ramanpreet Kaur Aujla

Diagonal Data Replication in Grid Environment 763
Rohaya Latip, Hamidah Ibrahim, Mohamed Othman, Md Nasir Sulaiman, and Azizol Abdullah

Efficient Shock-Capturing Numerical Schemes Using the Approach of Minimised Integrated Square Difference Error for Hyperbolic Conservation Laws 774
A.R. Appadu, M.Z. Dauhoo, and S.D.D.V. Rughooputh

Improvement on Real-Time Face Recognition Algorithm Using Representation of Face and Priority Order Matching 790
Tae Eun Kim, Chin Hyun Chung, and Jin Ok Kim

Modeling a Legged Robot for Visual Servoing 798
Zelmar Echegoyen, Alicia d'Anjou, and Manuel Graña

Information Extraction in a Set of Knowledge Using a Fuzzy Logic Based Intelligent Agent 811
Jorge Ropero, Ariel Gómez, Carlos León, and Alejandro Carrasco

Efficient Methods in Finding Aggregate Nearest Neighbor by Projection-Based Filtering 821
Yanmin Luo, Hanxiong Chen, Kazutaka Furuse, and Nobuo Ohbo

On Multicast Routing Based on Route Optimization in Network Mobility 834
Jong-Ki Kim, Kisoeb Park, and Moonseong Kim

An Effective XML-Based Sensor Data Stream Processing Middleware for Ubiquitous Service 844
Hun Soon Lee and Seung Il Jin

Opportunistic Transmission for Wireless Sensor Networks Under Delay Constraints 858
Ca Van Phan, Kikyung Baek, and Jeong Geun Kim

Workflow-Level Parameter Study Support for Production Grids 872
Peter Kacsuk, Zoltan Farkas, and Gabor Hermann

Certificate Issuing Using Proxy and Threshold Signatures in Self-initialized Ad Hoc Network 886
Jeonil Kang, DaeHun Nyang, Abedelaziz Mohaisen, Young-Geun Choi, and KoonSoon Kim

XWELL: A XML-Based Workflow Event Logging Mechanism and Language for Workflow Mining Systems 900
Min-Jae Park and Kwang-Hoon Kim

Workcase-Oriented Workflow Enactment Components for Very Large Scale Workflows 910
Jae-Kang Won and Kwang-Hoon Kim

A Workcase-Based Distributed Workflow Architecture and Its Implementation Using Enterprize Java Beans Framework 920
Hyung-Jin Ahn and Kwang-Hoon Kim

Building Web Application Fragments Using Presentation Framework ... 929
Junghwa Chae

Three–Dimensional Bursting Simulation on Two Parallel Systems 941
S. Tabik, L.F. Romero, E.M. Garzón, I. García, and J.I. Ramos

PAR Reduction Scheme for Efficient Detection of Side Information in OFDM-BLAST System 950
Myung-Sun Baek, Sang-Tea Kim, Young-Hwan You, and Hyoung-Kyu Song

Fuzzy PI Controller for Turbojet Engine of Unmanned Aircraft 958
Min Seok Jie, Eun Jong Mo, and Kang Woong Lee

Implementation of QoS-Aware Dynamic Multimedia Content Adaptation System 968
SooCheol Lee, DaeSub Yoon, Oh-Cheon Kwon, and EenJun Hwang

Experience of Efficient Data Transformation Solution for PCB Product Automation 978
Jung-Soo Han and Gui-Jung Kim

Performance Evaluation for Component Retrieval 987
Jung-Soo Han

The Clustering Algorithm of Design Pattern Using Object-Oriented Relationship 997
Gui-Jung Kim and Jung-Soo Han

Modeling Parametric Web Arc Weight Measurement 1007
Wookey Lee, Seung-Kil Lim, and Taesoo Lim

Performance Analysis of EPC Class-1 Generation-2 RFID Anti-collision Protocol 1017
Jeong Geun Kim, Woo Jin Shin, and Ji Ho Yoo

Worst-Case Evaluation of Flexible Solutions in Disjunctive Scheduling Problems 1027
Mohamed Ali Aloulou and Christian Artigues

The Search for a Good Lattice Augmentation Sequence in Three Dimensions 1037
Tiancheng Li and Ian Robinson

Tracing Illegal Users of Video: Reconsideration of Tree-Specific and Endbuyer-Specific Methods 1046
Hyunho Kang, Brian Kurkoski, Kazuhiko Yamaguchi, and Kingo Kobayashi

Rendering of Translucent Objects Based Upon PRT Techniques 1056
Zhang Jiawan, Gao Yang, Sun Jizhou, and Jin Zhou

An Image-Adaptive Semi-fragile Watermarking for Image Authentication and Tamper Detection 1066
Hengfu Yang, Xingming Sun, Bin Wang, and Zheng Qin

Identification of Fuzzy Set-Based Fuzzy Systems by Means of Data Granulation and Genetic Optimization 1076
Keon-Jun Park, Sung-Kwun Oh, Hyun-Ki Kim, Witold Pedrycz, and Seong-Whan Jang

Public Key Encryption with Keyword Search Based on K-Resilient IBE 1086
Dalia Khader

Efficient Partially Blind Signatures with Provable Security 1096
Qianhong Wu, Willy Susilo, Yi Mu, and Fanguo Zhang

Study on Grid-Based Special Remotely Sensed Data Processing Node ... 1106
Jianqin Wang, Yong Xue, Yincui Hu, Chaolin Wu, Jianping Guo, Lei Zheng, Ying Luo, RuiZhi Sun, GuangLi Liu, and YunLing Liu

Novel Algorithms for Quantum Simulation of $3D$ Atom-Diatom Reactive Scattering 1114
Ashot S. Gevorkyan, Gabriel G. Balint-Kurti, Alexander Bogdanov, and Gunnar Nyman

An Algorithm for Rendering Generalized Depth of Field Effects Based on Simulated Heat Diffusion 1124
Todd J. Kosloff and Brian A. Barsky

Fingerprint Template Protection Using Fuzzy Vault 1141
Daesung Moon, Sungju Lee, Seunghwan Jung, Yongwha Chung, Miae Park, and Okyeon Yi

Design and Application of Optimal Path Service System on Multi-level Road Network 1152
Yumin Chen, Jianya Gong, and Chenchen wu

Spatio-temporal Similarity Measure Algorithm for Moving Objects on Spatial Networks 1165
Jae-Woo Chang, Rabindra Bista, Young-Chang Kim, and Yong-Ki Kim

Efficient Text Detection in Color Images by Eliminating Reflectance Component 1179
Miyoung Choi and Hyungil Choi

Enhanced Non-disjoint Multi-path Source Routing Protocol for Wireless Ad-Hoc Networks 1187
Moon Jeong Kim, Dong Hoon Lee, and Young Ik Eom

Author Index 1197

Framework of Integrated System for the Innovation of Mold Manufacturing Through Process Integration and Collaboration

Bo Hyun Kim, Sung Bum Park, Gyu Bong Lee, and So Young Chung

Manufacturing Process Technology Division, Korea Institute of Industrial Technology
7-47, Songdo-dong, Yeonsu-gu, Incheon, Korea 406-840
{bhkim, sbred, gblee, motoya97}@kitech.re.kr

Abstract. The mold industry can be characterized as small businesses which hold only less than 10 to 20 people in most countries. A medium-size mold company has cooperated with its small suppliers to overcome this situation and to finish mold manufacturing processes from order to delivery. However, the collaboration in the mold value chain has not been preformed well because of the lack of the management transparency and manufacturing history, off-line communication of manufacturing information, etc. This study proposes a framework of the integrated system which can realize the concept of a mold value chain consisted of a medium mold company and its suppliers on the Internet. First of all, the current manufacturing process performed in the extended enterprise is analyzed using BPMN (business process modeling notation). The user requirements are collected by the interview with shop floor mangers and workers, and the major system functions are extracted from these requirements using KJ method. This study proposes the PPR (product/process/resource)-based architecture of the master data which plays a key role in the integrated database and the system framework consisted of six sub-systems. Finally, this study implements its prototype to check the validity of the system framework.

Keywords: collaboration, manufacturing process, BPMN (business process modeling notation), user requirement, master data, integrated system.

1 Introduction

The competitiveness of Korea mold industry can be summarized as the manufacturing technology based on high quality human resource [1]. However, it can be very difficult to continuously maintain the competitiveness using the human resource-based strategy because of the lack of skilled workers and the increase of labor cost. Furthermore, mold and die enterprises have been confronted with several difficulties such as the enlargement of the entrance into China, the decrease of investment, etc.

More than 90 percentages of Korea mold enterprises are small businesses which hold only less than 10 people [2]. Actually, a medium-size mold company, a leading

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1 – 10, 2007.
© Springer-Verlag Berlin Heidelberg 2007

company cooperates with some disjoint small suppliers to complete whole mold manufacturing processes from order to delivery. In other words, a mold value chain conceptually consists of the leading company and its disjoint suppliers as shown in Fig. 1.

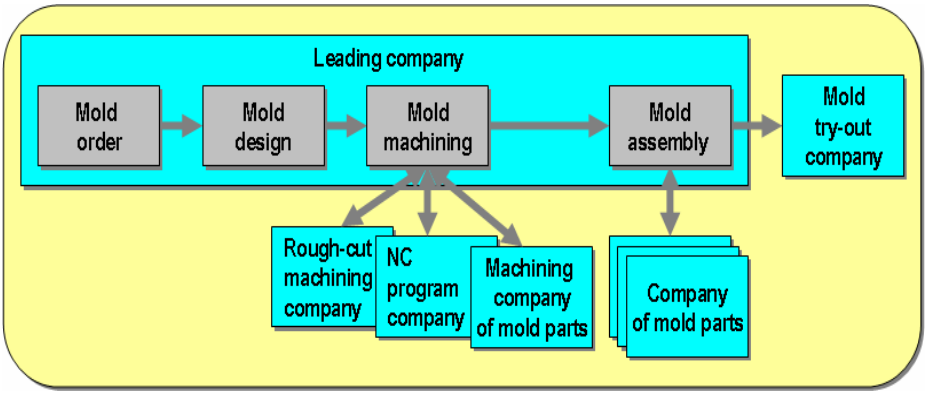


Fig. 1. The value chain of mold manufacturing

To reduce the delivery time and the manufacturing cost dramatically in the value chain, the leading company has made an effort to collaborate with its suppliers. However, the collaboration in the value chain has not been preformed well because of the lack of the management transparency and manufacturing history, off-line communication of manufacturing information, etc. [3]. This study proposes a framework of the integrated system which can realize the concept of an extended enterprise consisted of a medium mold company and several its suppliers on the Internet [4,5].

Before the design of the framework of the integrated system, this study analyzes the current business process of mold manufacturing and collects user requirements from managers and workers of the value chain. It also extracts the system functions from the user requirement using KJ method and establishes the to-be manufacturing process taking the usage of the collaboration system into account. Finally, it designs the overall system architecture and implements its prototype to check the validity of the system framework.

2 Analysis of Mold Manufacturing Process

A standard business process modeling notation (BPMN) will provide businesses with the capability of understanding their internal business procedures in a graphical notation and will give organizations the ability to communicate these procedures in a standard manner [6]. In this study, the BPMN is used to investigate the as-is process which includes the current mold manufacturing of the leading company and the interaction with its customers and suppliers.

As shown in Fig. 2, the top level as-is process model presents of the main processes performed by the leading company and the supporting processes performed by its suppliers. In the as-is process, the leading company receives mold orders from customers and starts to design molds. According to CAD data designed, workers generate machining data and cut mold bases using NC machines and EDM (electrode discharge machine). And, machined and standard parts delivered by suppliers are assembled into mold sets. After the try-out process for checking the manufacturing errors of molds, the QC (quality control) team holds a mold show in which all members related to manufacturing meet to evaluate the mold finally.

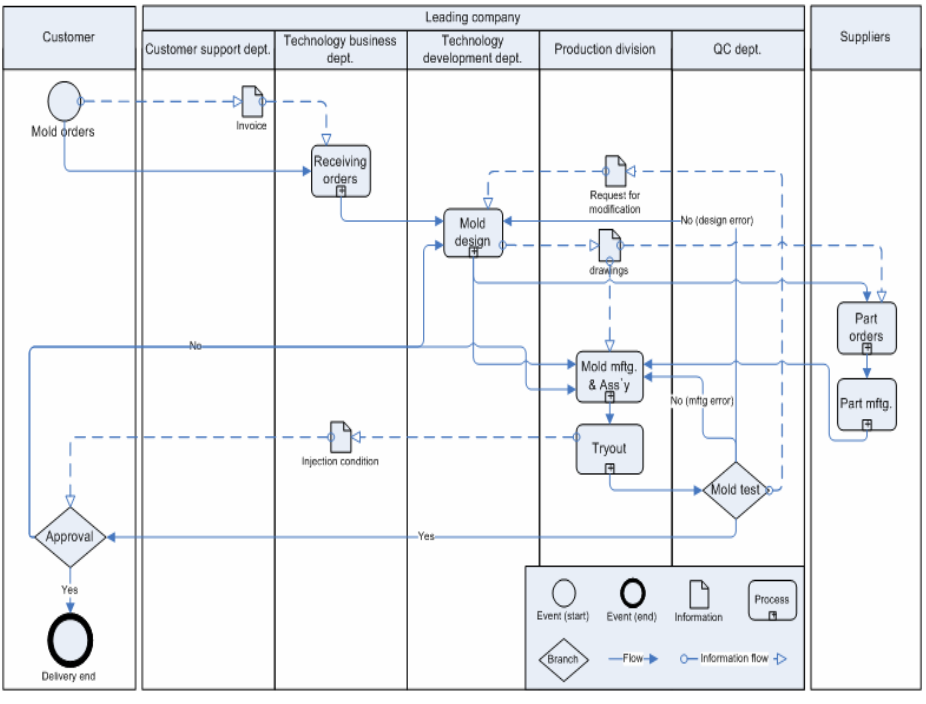


Fig. 2. Top level mold manufacturing process model

Using the final BOM (bill-of-material) and CAD data, the production control department establishes the process plan (PP) containing the manufacturing process sequence of molds and the daily production plan (DPP) consisted of the manufacturing schedule of molds and the work instruction of workers. According to the DPP, the purchase department gives its suppliers orders for materials, machined parts, and standard parts, and the NC and EDM department start to cut mold-bases and electrodes provided by its suppliers. Next, the mold-bases and core parts are moved into the polishing/assembly department for assembling mold sets.

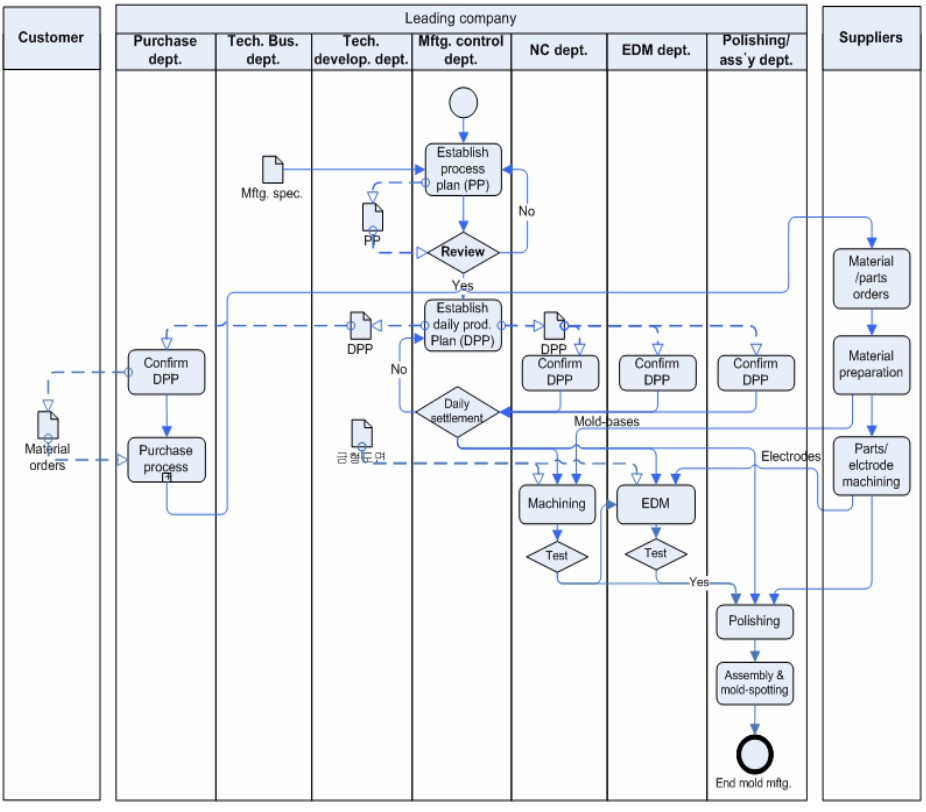


Fig. 3. Machining and assembly sub-process model

3 Extraction of System Functions

Before the design of the system functions, user requirements are collected from shop floor mangers and workers through the interviews and questionnaires as shown Table 1. Production and purchase department people, who frequently collaborate with their suppliers, require the control functions of manufacturing process and schedule, the purchase control functions related to outside machining and standard parts, etc.

The functions of the integrated system are extracted from the user requirements using KJ method, that is, a systematic classification method. Fig. 4 reveals that 22 major functions are defined from the requirements of each department, and have the complex relation among them.

For example, five major functions such as order control, customer support, turnkey manufacturing control, sales control, and mold history control are extracted from 22 requirements of the business department. In Fig. 4, the schedule control function of the production division plays a key role in managing the schedule of manufacturing process and has a close relation with the design schedule control function of the technology development department and the cost control function of the management

department. It provides the business department with the progress information of molds and sends the schedule of outside orders to the purchase department for controlling the purchase of standard parts, machined parts, and materials. The leading company can collaborate with its suppliers using shaded functions in Fig. 4.

Table 1. Summary of user requirements

Dept.	User requirements	Dept.	User requirements
Business dept.	Analysis of order validity	Production division	Process plan of outside order
	Order registration		Control of standard time
	Order plan and status monitoring		Work plan and assignment
	Control of estimate history		Inquiry of work result and status
	Development status report		Work instruction
	Comparison progress with result		Work result
	Control of customer data		Monitoring of operation status
	Decision support for turnkey orders		Analysis of non-operation time
	Progress control of turnkey orders		Maintenance plan
	Supply control	Purchase dept.	Control of registration status of outside machining companies
	Status control of bill collecting		Decision support of inside or outside making
	Sales plan/result control		Control of progress status of outside machining
	History of design change		Control of outside machining cost
	History of transfer/repair		Order generation
Technology development dept.	Establishment of design schedule per mold		Control order status per company
	Inquiry of design schedule and result per mold		Control of estimate/detailed transaction
	Inquiry of design progress status per customer		Control of purchase status per mold
	Model registration		Entering control
	Model search		stock control
	Customer data store control		Control of facility history
	Registration of standards part code		Control of repair details
	Search/update of standard part		Control of unit price of standard part
	Control of standard part unit price	Management dept.	Control of production status
	Customers' ECO control		Control of sales status
	Design change control in the company		Control of order status
	Drawing transfer of outside machining		Cost analysis per mold/process
	Drawing transfer in the company		Overtime plan
	Input and inquiry of BOM		Extra work plan
Production division	Establishment of main schedule	IT department	User registration
	Inquiry and report of main schedule per customer and mold		Authority control of user and organization
	Control of outside order schedule		Control of code of standard facility, process, and parts
	Establishment of process plan		Control of calendar, capacity, and unit labor cost
	Control of schedule per process		Control of interaction with groupware and homepage
	Control of urgent process change		Security control of system

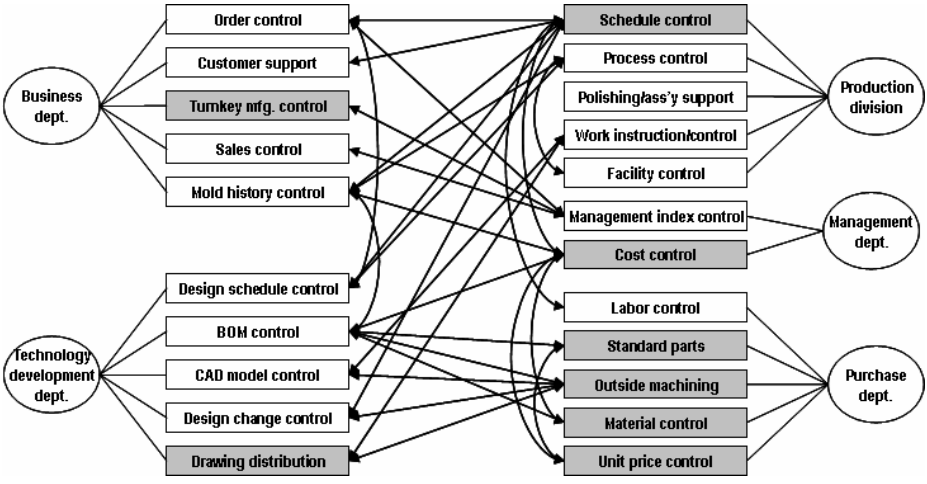


Fig. 4. Extracted system function and their relations

4 Design of System Framework

Mold companies of middle standing in Korea mostly have the legacy information system such as ERP (enterprise resource planning), PDM (product data management) system, etc. Before the design of the integrated system, thus, we should analyze the legacy systems and design the integrated database which can cover the legacy systems and the collaboration system. Especially in the design of the integrated database, the interface among these information systems should be considered.

The master data plays an important role in linking the database of the legacy systems and that of the collaboration system. This study proposes the PPR (product/process/ resource)-based architecture in the design of master data. In the PPR-based architecture of Fig. 5, the product data includes molds specification, BOP (bill-of-process) containing BOM and process, part information, and customer information, the process data contains the information of inbound process, suppliers' process, and workers, and the resource data consists of facility specification, characteristics, and history. Let us assume that the product, process, and resource data are located on the x-, y-, and z-axis of 3-dimensional space respectively. Then, the product-process, product-resource, and process-resource data can be explained as the xy-, xz-, and yz-plane respectively. In other words, a complex data connecting to the information systems can be divided into PPR-based master data.

Form the define system functions, the PPR-based architecture of master data, and the information the legacy systems, this study proposes the overall system framework as shown in Fig. 6. The integrated system consists of the collaboration portal, the collaboration service system, the supplier control system, the MES (manufacturing execution system), the flow control system of technology information, and the legacy system.

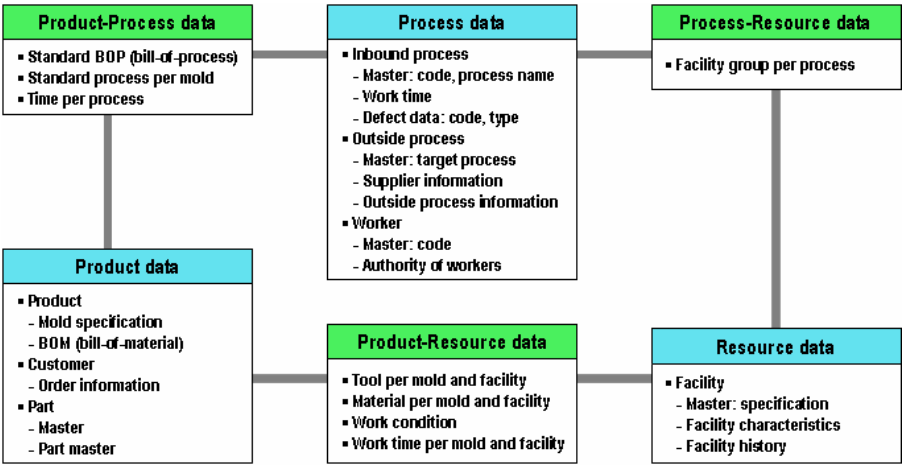


Fig. 5. PPR-based architecture of master data

The collaboration portal provides managers, customers, and suppliers with the key performance index of mold shops, the due date and progress of requested molds, and the order and technical information respectively. The supplier control system contains the internal system functions to manage the outside orders of the material, machine parts, and standard parts and the connection function of the suppliers' facility to monitor their status. Suppliers can share CAD data and hold CAD conference, and inquire the schedule of mold projects in which they have joined actually through the collaboration service system. The MES system is in charge of the major functions of the production division such as the function of the schedule control, the process control, the polishing/assembly support, the work instruction/control, and the facility control as shown in Fig. 4. In other words, the MES system plays a key role in the inbound production control and interacts with the supplier control system, the collaboration service system, and the legacy systems. The flow control system of technology information gives workers work instructions, and receives the actual results of workers and the status information of inbound facilities directly connected to it.

5 Prototype of System

The current manufacturing process and environment should be changed and redesigned to employ the integrated system. To construct the system systematically, this study proposes the task force team consisted of all departments which will use the integrated system. The role of the task force team is to establish the to-be process and to propose their know-how and idea about the integrated system. Fig. 7 shows a part of the to-be process which describes the business process of the purchase department and suppliers, and defines the system functions to be used.

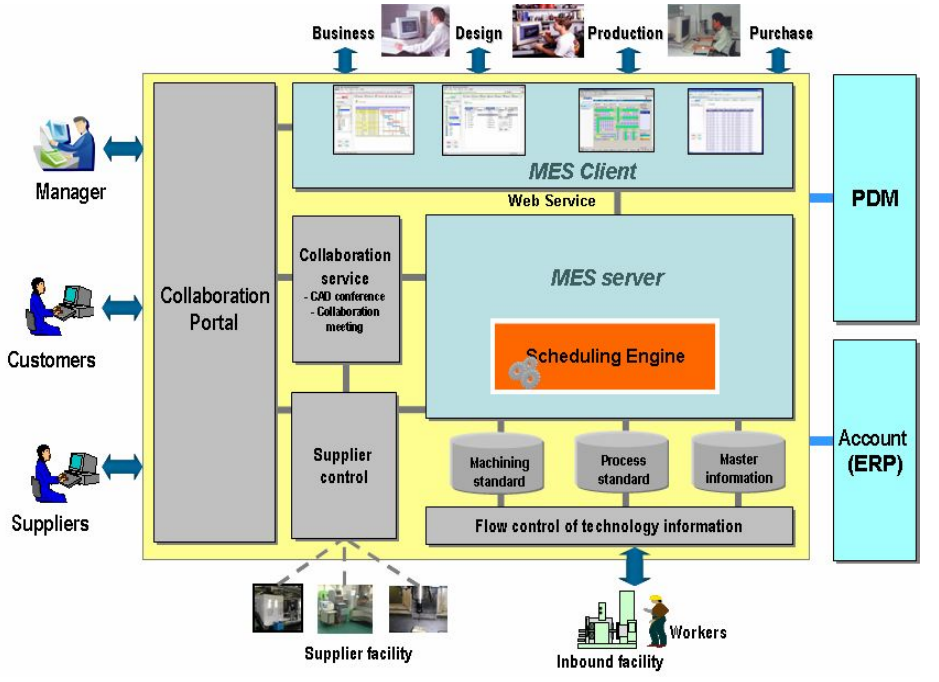


Fig. 6. Framework of integrated system

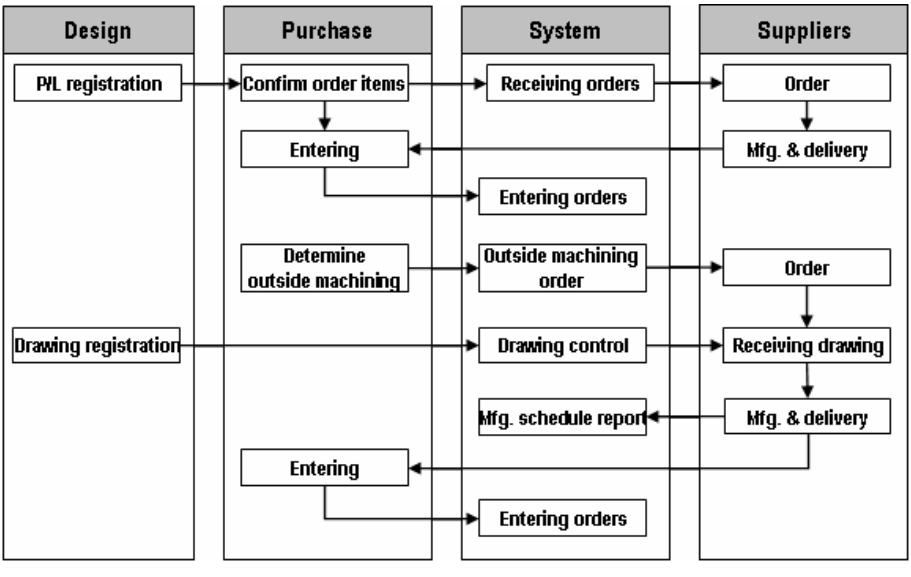


Fig. 7. A part of the to-be process in the purchase division

This study implements a prototype of the integrated system according to the proposed system framework. Fig. 8 shows a part of users' scenario and example windows in the business process of the purchase department. Users can check up the list of material orders using the upper window of Fig. 8 and edit the outside orders in detail using the lower window of Fig. 8.

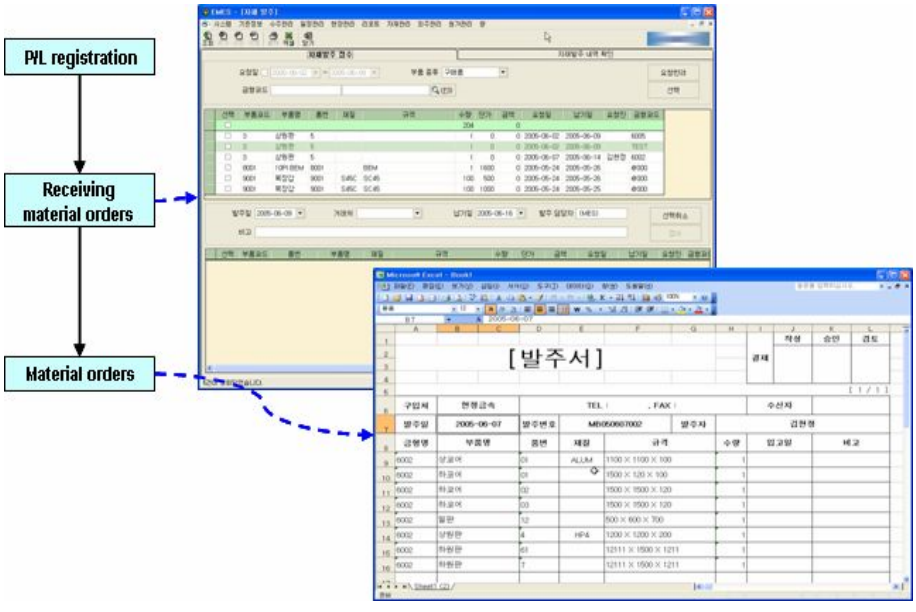


Fig. 8. Examples windows of a prototype system

6 Conclusion

This study proposes a framework of the integrated system which can realize the concept of an extended enterprise consisted of a medium mold company and several its suppliers on the Internet. At first, the current manufacturing process performed in the extended enterprise is analyzed using BPMN (business process modeling notation). To develop the user-centric system, this study collects the 68 user requirements from shop floor mangers and workers through the interview and questionnaire and extracts the 22 major system functions from these requirements.

As the first step of the system design, this study establishes the integrated database which covers all manufacturing information system in the mold value chain. The PPR (product/process/resource)-based architecture is proposed in the design of master data which plays an important role in the integrated database. This study also proposes the system framework consisted of six sub systems: the collaboration portal, the collaboration service system, the supplier control system, the MES (manufacturing execution system), the flow control system of technology information, and the legacy system.

Next, the to-be manufacturing process is designed taking the usage of the collaboration system into account. To construct the integrated system in a short period, this

study proposes a special team consisted of key man of all departments. The role of the special team is to confirm the to-be process designed and to propose their know-how and idea about the integrated system. A prototype of the integrated system is implemented according to the proposed system framework. The integrated system should be applied to real mold value chains to check its validity.

References

1. Choi, B.K.: B2B pilot project for the die/mould industry in Korea. Asia e-Biz workshop, Beijing, China (2001)
2. KITECH and KDMIC: Report on the survey of technology and management environment of mould and die shops in Korea (2002)
3. Kim, B.H., Ko, K.H., Choi, B.K.: Technical Issues in the Construction of Intelligent Mould Shop. In: IMS International Forum 2004, Como, Italy, pp. 62–69 (May 2004)
4. e-Manufacturing Project Team, KITECH: New Innovation of Manufacturing Industry, e-Manufacturing Project, pp. 136-144 (December 2005)
5. KITECH: International Forum on Manufacturing Innovation. (2005), http://mif.i-mfg.com
6. BPMI.org: Business Process Modeling Notation (BPMN), Version 1.0 – (May 2004)

A Study on Automated Design System for a Blow Mould

Yong Ju Cho, Kwang Yeol Ryu, and Seok Woo Lee

i-Manufacturing Center, Korea Institute of Industrial Technology,
7-47, Songdo-dong, Yeonsu-gu, Incheon Metropolitan City 406-840, Korea
{yjcho, ducksal, swlee}@kitech.re.kr

Abstract. Designing a blow mould is a very complicated and knowledge-intensive process. This paper describes a feature-based, parametric-based, and knowledge-based design system for drawing blow moulds that requires only a minimum set of parameters to be set before it completes the design of the main parts of an injection, blow, and eject mould. This blow mould design system, implemented on top of the Powershape CAD, consists of a machine type and cavity setting module, an injection designer, a blow designer, an eject designer, mould design creation module, moulding database, and machine database. Through these modules, the mould designers are able to create a 3D blow mould design from customer requirements. This system is capable of reducing the lead-time of blow mould design as it integrates the three stages (injection, blow, eject) of the blow mould design process and can greatly improve the product quality through a standardization of the design process.

Keywords: Blow mould design, Knowledge-based system, Parametric-based system, Computer-Aided Design.

1 Introduction

Blow moulding is the world's third largest plastic processing technique and is used to produce hollow, thin-wall objects from thermoplastic materials [1]. Over the past 20 years, blow moulding has undergone rapid growth with the development of new application areas in the automotive, sports and leisure, electronics, transportation, office automation equipment, and packaging industries [2][3]. In general, there are two major types of blow moulding; Extrusion Blow Moulding (EBM) and Stretch Blow Moulding (SBM). The former is widely used to produce containers of various sizes and shapes and is also adapted to make irregular complex hollow parts, such as those supplied to the automobile, office automation equipment and pharmaceutical sectors, etc. The latter technique is extensively used in the commercial production of bottles for food, beverages, and pharmaceutical industries [4].

Blow products go through the design, machining, and injection processes and the designing of a blow mould is a vital stage in the development of blow products. In addition to the highly complex blow mould structures, interferences among various design parameters make the design task extremely difficult. Therefore, majority of people still believe it is art rather than science that plays an important role in the mould designing [5]. In general, it requires much time and effort to become a mould design expert.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 11 – 19, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The highly competitive environment makes it necessary to reduce the time and money spent on designing moulds while maintaining high standards for product quality. Therefore, the use of computer-aided design (CAD) has become one of the most important way to increase productivity. However, the 2D design systems for designing a blow mould possess many problems as shown below:

① Difficulty of understanding the dimensions and configuration in the drawing
Difficulty of checking interferences
Difficulty of designing to the specifications of individual injection machines
Blow mould non-experts' inability to work
Inability to calculate the weight of blow products
⑥ Occurrence of unnecessary and simple repetitive work
Difficulty of standardizing the mould
Difficulty of redrawing

These drawbacks of the traditional 2D design systems make them very time-consuming to learn and use. In the end, such drawbacks would affect product delivery and ultimately lead to deterioration in quality. With a desire to maintain the competitive edge, mould-making companies shorten both design and manufacturing lead-time and improve product quality through 3D computer-aided design (CAD) system. The 3D CAD system uses intuitive and concrete methods to represent objects to its users by resolving the problems of the traditional 2D design systems.

However, most 3D CAD software programs offer only the simple geometrical modeling function and fail to offer sufficient design knowledge to users. Therefore, the design of automatic, knowledge-based, and intelligent systems has been an active research topic for a long time [5]. Many researches have been conducted to apply these systems to mould design (Table 1), but no system has been developed for applying to the blow mould design.

In order to complete the blow mould design process in a scientific approach, this paper introduces a feature-based, parametric-based and knowledge-based design

Table 1. Related researches for mould design

Design Methodology	Areas	Environments	Related works
Feature-based parametric design	Gating system for a die-casting	Unigraphics CAD	[6]
-	Entire die design	AutoCAD	[7]
Semi-automated	Injection mould design	-	[8] IMOULD (intelligent mould design and assembly system)
Parametric modeling	Set-based design	-	[9]
-	Plastic injection mould design	Solid Works	[10]
Parametric design	Tire mould	CATIA	[11]
Knowledge-based	Stamping tool design	AutoCAD	[12]

system for designing a blow mould, where designers only need to input a minimum set of design parameters, and the system will automatically complete the design of the injection part, blow part, and eject part of the blow mould. Moreover, the BMM (Blow Mould Maker) system proposed in this study ensures that, when the designer modifies the mould design, these changes can be easily performed. The system is developed using Visual Basic .net language on the Powershape 7.0 platform. The detailed contents of the work are described in the following sections.

2 Procedure of Making Blow Molding

2.1 SBM Introduction

Blow moulding refers to the process of preprocessing thermoplastic plastics in a mould called parison, blowing in air for fitting, and making it adhere closely to the inner wall of the mould. In general, there are two main types of blow moulding; Extrusion Blow Moulding (EBM) and Stretch Blow Moulding (SBM). This paper focuses on the stretch blow moulding. In the Stretch Blow Moulding (SBM) process, the plastic is first moulded into a "preform" using the injection mould process. These preforms are produced with the necks of the bottles, including threads (the "finish") on one end. These preforms are packaged, and fed later (after cooling) into an EBM blow moulding machine. In the SBM process, the preforms are heated (typically using infrared heaters) above their glass transition temperature, and then blown using high-pressure air into bottles using metal blow moulds. Usually the preform is stretched with a core rod as a part of the process. This process is illustrated in Figure 1.

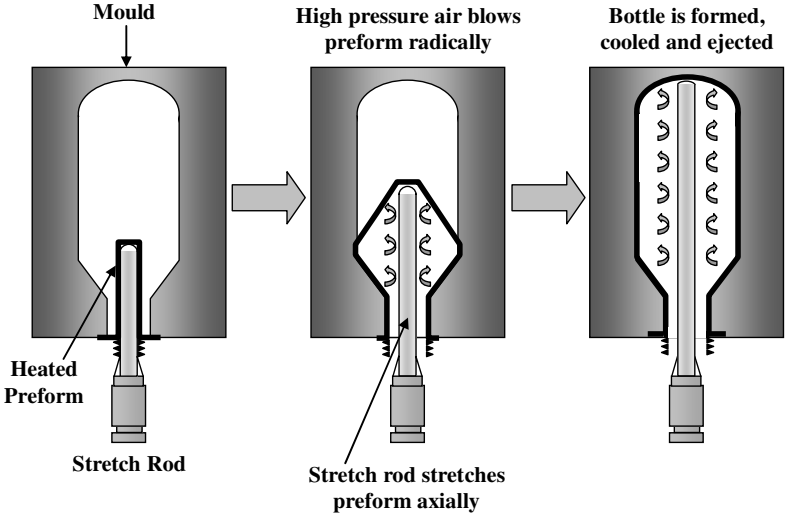


Fig. 1. Stretch-blow moulding of PET bottles ([13])

Additionally, the SBM machine can be divided into the 3 stroke (injection-blowing-ejection)-type machine and the 4 stroke (injection-heating-blowing-ejection)-type machine. In general, the 3-stroke-type machine (ex. AOKI machine) is used to make small products, whereas the 4-stroke-type machine (ex. ASB machine) is used to make large products.

2.2 Business Process for Blow Mould Production

The process of making a blow mould consists of 5 stages: Ordering, Pre-examination of customer requirements, Contract, Design of mould, and Machining of mould and Pre-injection. The details of each stage are described below:

1) Ordering
A blow mould company takes orders from customers through a request for manufacturing that includes the injection moulding machine type, product weight, product capacity, neck specifications, number of cavities and time limit for delivery, and receives information on the rough configuration and concept of the product. Customers deliver a rough concept of the desired product since they cannot be certain about the formability of the product.

2) Pre-Examination of Customer Requirements
Preliminary review of customer requirements is a preparatory stage for receiving orders for moulds, thus it is important to reflect customer requirements as much as possible so as to quickly understand the formability of the ordered product. At this stage, the blow mould company performs three major tasks. First, the company takes advantage of the information contained in the customer's request for manufacturing and product concept to perform the 2D product drawing work and the modeling task based on the 3D CAD. Second, the blow mould company performs the 2D mould design reflecting customer requirements and the specifications of the injection machine in order to verify product formability. Third, the company checks the detailed machine specifications required by the customer.

3) Contract
The blow mould company utilizes the 2D drawing and 3D modeling of the product and the mould drawing to reach an agreement with the customer. Lastly, if the customer confirms product specifications, the contract will be concluded.

4) Design of the Mould
First, the mould company uses the AutoCAD and CADKEY design system to work on the drawing that includes the neck specifications, the length, and design of the preform. The preform design is quite a complicated process, and a very sensitive part of the moulding. What follows next is the design of the mould based on the detailed specifications of the preform. In this stage, the mould drawing for a 3-stroke machine or 4-stroke machine is designed for each stage respectively. If the preform and mould design are completed, the mould company orders necessary materials according to the

BOM (Bill of Materials). Lastly, the company delivers the preform drawing and mould drawing to the production department for the mould production.

5) Machining of the Mould and Pre-injection

The actual mould is produced based on the finalized preform drawing and mould drawing. Processing largely consists of lathe, milling, grinding and assembly stages. Next is the evaluation of overall quality of the assembled mould before delivering it to the customer. Then lastly, pre-injection is performed as a final test. This stage checks whether products are made according to the customer requirements.

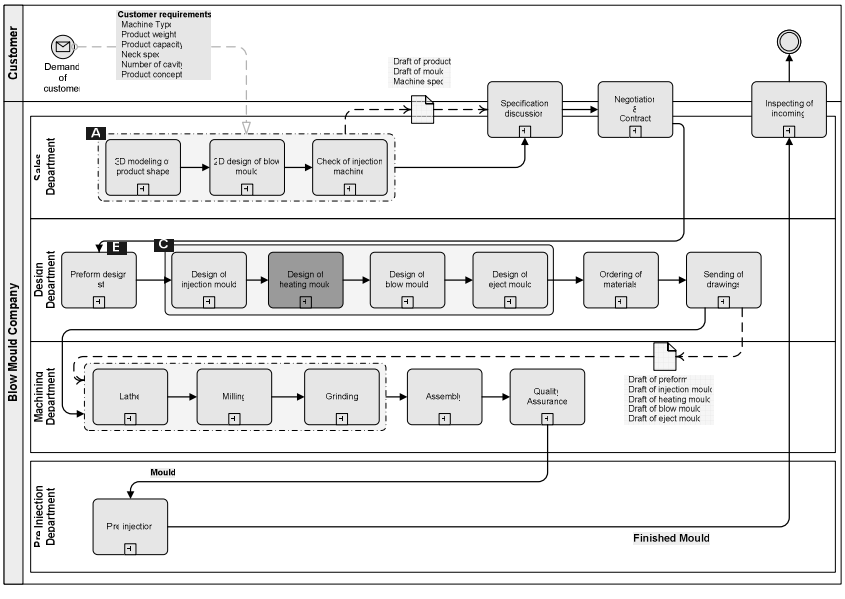


Fig. 2. Process of blow mould making

We have illustrated the analysis result of the blow mould manufacturing process in Figure 2 by utilizing BPMN (Business Process Modeling Notation). Today, the blow mould industry is shifting to small quantity batch production and customers are demanding ever quicker delivery. In general, customers demand 30-day limit for delivery, but the net working time of the blow mould company is only about 20 days or so excluding weekends. In Figure 2, the design area is A, B, and C, and ordinarily, the blow mould company spends about 1-2 days for job A, and 4-5 days for jobs B and C. On the other hand, the number of working days for the mould machining job is almost fixed since this job is performed according to a prefixed schedule. Such a reality clearly demonstrates the need for improving the design work in order to shorten the production time needed to make a blow mould. The BMM (Blow Mould Maker) system can be applied to design areas A and C. However, BMM cannot be applied to the preform design work in area B since this area is too complicated for automation.

3 Design Approach

The design approach undertaken in this work includes (1) feature-based modeling (2) parametric-based design, and (3) knowledge-based design.

First, feature-based modeling constructs the model of the product directly on the basis of feature [5]. In this study, standard parts whose dimensions are seldom changed are registered in the component library of Powershape, and BMM calls necessary parts for blow mould design. In addition, if certain features are necessary, such can be added to the component library. By using this way the designer can minimize repetitive tasks.

Second, parametric-based design deals with variable dimensions as control parameters and it is an efficient tool for creating models based on parameters [5]. Parameters used in the BMM system are largely divided into two categories. First is the parameters inputted and defined directly by the designer, and second is the parameters calculated automatically based on the values inputted by the designer. The designer has no control over the latter parameters. Parametric design not only increases the design efficiency, but also makes the updates and modifications of existing designs easier and faster since these can be achieved by changing the parameters of the parametric model [14][15].

Third, knowledge is a set of rules derived from the designer's knowledge and know-how, and can be considered as a type of design guide for applying the best design know-how. The design know-how unique to the blow mould industry is implemented in the knowledge in the BMM system proposed in this study. Generally, in the blow mould industry, mould design is demanded according to the specifications and mechanical characteristics of the injection machine owned by the customer. To this end, mould design specifications that correspond to 3-stroke machines (AOKI-100L, AOKI-250LL-50, AOKI-250LL-50S, AOKI-350LL-40, AOKI-500LL-75) and 4-stroke machines (ASB-50H, ASB-70DPH, ASB-650NH) are incorporated into the BMM system (Table 2). Also, transforming the design work into knowledge is related to the preform design. The preform design task cannot be automated in the BMM system, and is the most important factor that affects the quality of blow products. In connection with this, the existing preform 2D drawings within the blow mould company were databased and transformed into knowledge. The designer takes advantage of the existing preform 2D drawings to execute the best mould design by means of the BMM system.

4 System Implementation

A prototype of the mould design system for a blow mould has been implemented using a PentiumTM IV PC-compatible as the hardware. This prototype system uses a commercial CAD system (Powershape 7.0) and a commercial database system (Microsoft AccessTM) as the software. The prototype system is developed using the Microsoft Visual Basic .Net programming language and Powershape API (Application Programming Interface) in a Windows XPTM environment. Powershape provides a user-friendly API. Using this API, an intelligent and interactive design environment for blow mould design is developed. There are many useful menus, tool buttons, dialogues and commands.

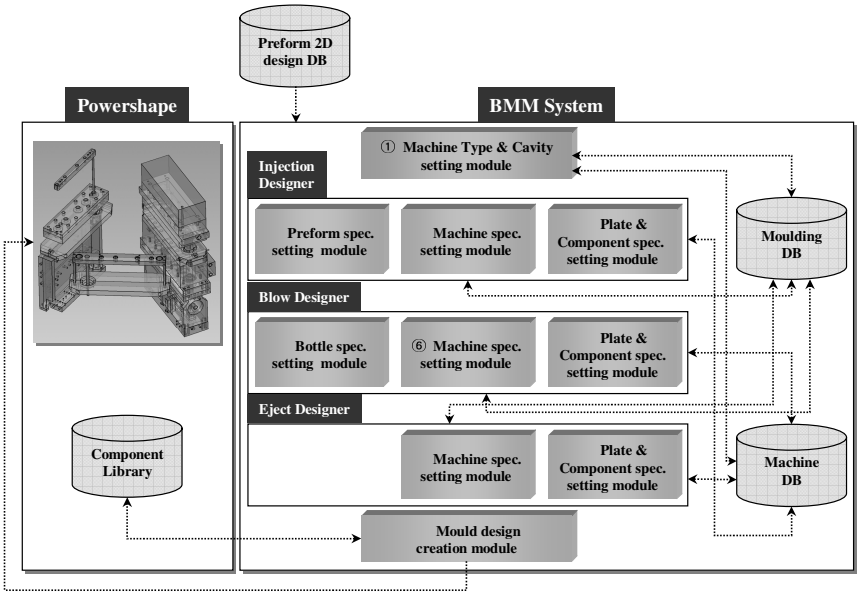


Fig. 3. BMM system architecture

The proposed BMM system consists of 10 modules (Figure 3). Of the 10 modules, Modules 2, 3 and 4 are grouped for the injection designer, Modules 5, 6, and 7 are grouped for the blow designer, and 8 and 9 are grouped for the eject designer according to mould types. And, the BMM system has a moulding database and a machine database. The moulding database defines the types of machines, and the number of cavities each machine can have. It also holds the database for each of the eight machines defined in the moulding database, and each database defines the design information on all parts contained in each machine.

5 Conclusion

This paper presented a feature-based, parametric-based, and knowledge-based design system for drawing the blow mould, which was built on top of the Powershape CAD software. Unlike the injection mould, the blow mould must go through the 3-stage process, i.e. injection, blowing and ejection, to make the final product, and separate moulds must be used in different stages. The BMM system helps the designer to design the three moulds easily and quickly. The advantages of the developed blow mould design system are as follows:

The BMM system has user-friendly interfaces.

Since it makes use of databases, it is highly flexible, and the blow mould industries that have their own standards can customize the databases to suit their needs.

The BMM system is capable of reducing the lead-time of blow mould design as it integrates the three stages (injection, blow, eject) of the blow mould design process and can greatly improve the design quality through a standardization of design process.

This system enables blow mould designers to have more useful technical discussions prior to blow mould machining as changes to the mould design can be made immediately during the meeting.

However, the developed system has some limitations. The BMM system developed in this study is a mould design system for making PET bottles in the 1 stage machines. In general, since the blow mould industry is considerably dependent on the injection machine, a mould design system for 2 stage machines, which are given a great deal of weight in a blow mould company, must be developed and integrated.

References

1. Tahboub, K.K., Rawabdeh, I.A.: A design of experiments approach for optimizing an extrusion blow molding process. Journal of Quality in Maintenance Engineering 10(1), 47–54 (2004)
2. Gao, D.M., Nguyen, K.T., Hetu, J.F., Laroche, D., Garcia-Rejon, A.: Modeling of industrial polymer processes: injection molding and blow molding. Advanced Performance Materials 5(1-2), 43–64 (1998)
3. Huang, H.X., Liao, C.M.: Prediction of parison swell in plastics extrusion blow molding using a neural network method. Polymer Testing 21(7), 745–749 (2002)
4. Huang, H.X., Li, Y.Z., Deng, Y.H.: Online real-time acquisition for transient temperature in blow molding. Polymer Testing 25, 839–845 (2006)
5. Lin, B.T., Chan, C.K., Wang, J.C.: A knowledge-based parametric design system for drawing dies. International Journal of Advanced Manufacturing Technology DOI:10.1007/s00170-006-0882-y (2007)
6. Wu, S.H., Lee, K.S., Fuh, J.Y.: Feature-based parametric design of a gating system for a die-casting die. International Journal of Advanced Manufacturing Technology 19, 821–829 (2002)
7. Choi, J.C., Kwon, T.H., Park, J.H., Kim, J.H., Kim, C.H.: A study on development of a die design system for die-casting. International Journal of Advanced Manufacturing Technology 20, 1–8 (2002)
8. Lee, K.S., Fuh, J.Y.H., Zhang, Y.F., Nee, A.Y.C., Li, Z.: IMOLD: An intelligent plastic injection mold design and assembly system. In: Proceedings of the 4th international conference on die and mold technology, Kuala Lumpur, Malaysia, pp. 30–37 (1997)
9. Nahm, Y.E., Ishikawa, H.: A new 3D-CAD system for set-based parametric design. International Journal of Advanced Manufacturing Technology 29, 137–150 (2006)
10. Kong, L., Fuh, J.Y.H., Lee, K.S., Liu, X.L., Ling, L.S., Zhang, Y.F., Nee, A.Y.C.: A Windows-native 3D plastic injection mold design system. Journal of Materials Processing Technology 139, 81–89 (2003)
11. Chu, C.H., Song, M.C., Luo, C.S.: Computer-aided parametric design for 3D tire mold production. Computers in Industry 57, 11–25 (2006)
12. Cheok, B.T., Nee, A.Y.C.: Trends and developments in the automation of design and manufacture of dies for metal stampings. Journal of Materials Processing Technology 75, 240–252 (1998)

13. Yang, Z.J., Harkin-Jones, E., Menary, G.H., Armstrong, C.G.: Coupled temperature-displacement modeling of injection stretch-blow moulding of PET bottles using Buckley model. Journal of Materials Processing Technology 153–154, 20-27 (2004)
14. Roller, D.: An approach to computer-aided parametric design. Computer Aided Design 23(5), 385–391 (1991)
15. Anderl, R.: Parametrics for product modeling; in Parametric and Variational Design. Hoschek I, Dankwort, W., (eds.), Teubner Verlag (1994)

Development of an Evaluation System of the Informatization Level for the Mould Companies in Korea

Yong Ju Cho and Sung Hee Lee

Precision Molds & Dies Team, Korea Institute of Industrial Technology,
7-47, Songdo-dong, Yeonsu-gu, Incheon Metropolitan City 406-840, Korea
{yjcho, birdlee}@kitech.re.kr

Abstract. Currently, the domestic mould industry is experiencing tremendous difficulty in promoting informatization due to a lack of confidence in the effect of informatization and the related shortage of manpower and funding. For instance, they have difficulty in selecting the type of information system they should introduce in the future. In this study, we develop a model for evaluating informatization level of the mould companies and the evaluation framework from the information viewpoint and derive three evaluation fields and their respective factors as well as a questionnaire for each factor to conduct a survey of domestic mould companies. After the reliability and validity of the survey were analyzed, we finalize the evaluation framework.

Keywords: Manufacturing, Evaluation System, Information Technology, Measurement, Survey.

1 Introduction

From a historical viewpoint, enterprise informatization could be cited as a social system that appeared after the advent of the industrial society. Accordingly, we should evaluate enterprise informatization from a dynamic viewpoint of not only simply the change in overall corporate business from the development of information technology but also the change in the enterprise resulting from the fluctuation in the corporate environment.

The objectives of this study are as follows. First, we suggest a model to evaluate the informatization level of mould companies by taking into account three fields, information/network infrastructure, mould manufacturing information system, and collaboration. Second, we verify the suggested evaluation model through a survey. Third, we present methods to effectively cope with the demands on informatization faced by domestic mould companies.

2 ESMI (Evaluation System of Mould Industry Informatization)

A model for evaluating the informatization level of the mould industry is aimed at efficient propulsion of informatization as a means of securing the competitiveness of the mould industry. To precisely evaluate the informatization level of an enterprise,

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 20 – 32, 2007.
© Springer-Verlag Berlin Heidelberg 2007

the level must be analyzed from business and information viewpoints. Generally, because the information in the manufacturing industry flows along the business process, information and business must be considered concurrently. First, from the business viewpoint, five aspects should be considered in the evaluation of informatization level. Although the five elements are classified into separated items, they can be considered parts of a workflow series at the mould companies. For example, in business, design, and production departments, outputs are yielded in the departments' business process. The outputs are documents, drawings, parts and moulds. Such activities are taking place in relation with the company's profits. In this way, all four factors are involved in a continuous process closely related to the factors that constitute the information view. Likewise, customer-related businesses of the mould companies should be considered from the business view. In mould companies, most of the businesses are conducted with many customers. Second, from the information viewpoint, three fields (information/network infrastructure, mould manufacturing information system, collaboration) should be considered in the evaluation of informatization level. The information/network infrastructure field is composed of hardware or software, which are the basic means and manpower in enterprises. Recently, advanced IT technology such as wireless LAN has been adopted by manufacturing enterprises. As such, network infrastructure items, which can reflect the state-of-the-art Internet and network technology, should be included in information/network infrastructure field. MMIS (Mould Manufacturing Information System) field is composed of the information systems that can reflect the business characteristics of the mould industry; namely, the businesses of the mould industry could be divided into three areas (business, design, and production). Accordingly, the MMIS field should support the informatization of the three types of businesses. The collaboration field reflects the collaboration within an enterprise and the collaboration between enterprises; it also includes the business process for collaboration and information system supporting collaboration.

We would like to suggest the ESMI model (Fig. 1), an evaluation model of informatization level, which can be applied to the mould industry from the viewpoints of information and business. The empty boxes in Figure 1 imply that the items from the information viewpoint can be added, deleted or integrated. Eleven items from the information viewpoint can be classified as follows: information/network infrastructure, mould manufacturing information system, and collaboration field as mentioned above. We classified the information/network infrastructure into two fields: information facility and human resource.

The information/network infrastructure area can be seen as the foundation layer for promoting informatization in the mould industry. The scope of the information facility includes hardware, software, web utilization and network-related items. Meanwhile, human resource includes manpower-related items and informatization effect items. The information system, as mentioned above, is mapped to the area of a mould manufacturing information system unique to the business of the mould industry. This area is the second layer, which can support and manage sales, design and production performed in the mould industry by utilizing an information/network infrastructure. The third layer includes the internal and external collaboration within the mould industry and is mapped to the area of collaboration related with information.

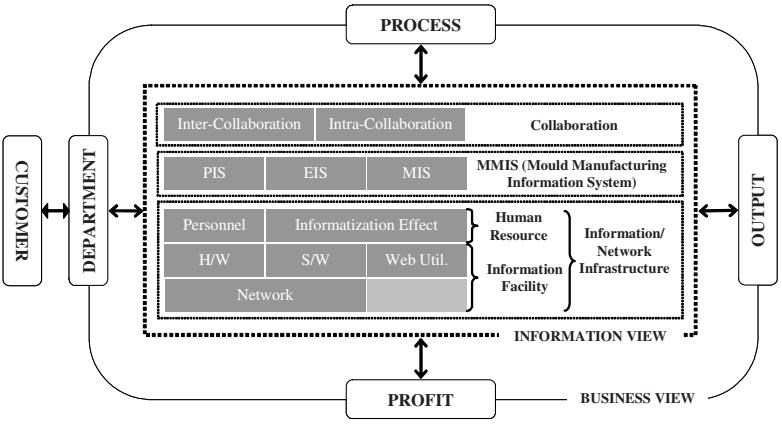


Fig. 1. ESMI model

Figure 2 shows the overall framework of the ESMI model, developed in this paper, for evaluating the informatization level of mould enterprises from the information viewpoint. The evaluation fields were largely classified as follows: information/network infrastructure, mould manufacturing information system, and collaboration. The information/network infrastructure field is regarded as a common field in the evaluation of the overall informatization of mould enterprises; the mould manufacturing information system and collaboration are evaluation fields reflecting the characteristics of the mould industry.

In this paper, the information/network infrastructure field to measure information facility and human resource was largely divided into two fields: technical component and personnel component. Further, the mould manufacturing information system field

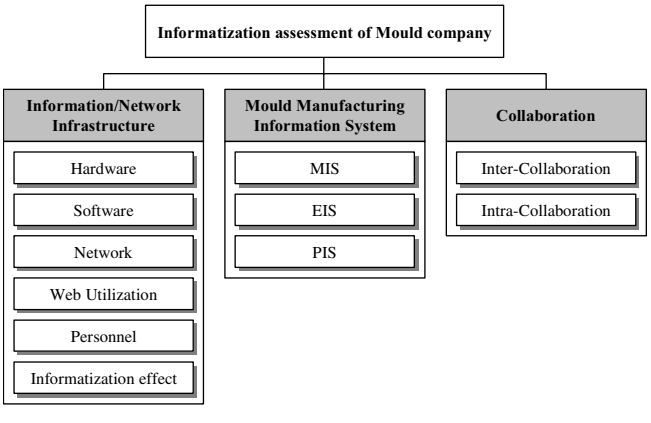


Fig. 2. Evaluation framework of the ESMI model (1st)

was divided into MIS (Management Information System), EIS (Engineering Information System), and PIS (Production Information System) by analysing the business of the mould industry. Collaboration was divided into inter-collaboration within mould companies and intra-collaboration between mould companies. The next chapter explains the details of the three evaluation fields.

2.1 Information/Network Infrastructure Level

Information technology is an important competitive edge in the manufacturing industry [1, 2, 3]. Moreover, information infrastructure is a system using computers and communication for acquisition, storage, processing, delivery, and use of information. It also supports the establishment of a management strategy and is regarded as the management foundation offering convenience and effectiveness in business activities to members of an enterprise. To attain a sustainable competition advantage, Davenport and Linder suggested the focus on information technology infrastructure, rather than on application [4]. Also, information technology infrastructure is a major management resource, being one of the few sources that can secure superiority in competition. As well, the cost for investment in information technology infrastructure accounted for 58% of the information technology budget of an enterprise.

In this paper, we classified the information/network infrastructure into two components: technical component and personnel component (Table 1). The technical component consists of the physical asset and the personnel component consists of the intellectual asset. Such factors as hardware, software, network, and web utilization are all considered information facility and were drawn from physical asset. The personnel and informatization effect were drawn from the intellectual asset. Table 1 elucidates the factors and items in the information/network infrastructure field. That is, a total of six factors were extracted as factors of the information/network infrastructure, as well as evaluation items related to each factor.

Table 1. Classification of the Information/Network Infrastructure

Field	Category	Factors	Items
Information/Network Infrastructure	Technical Component (Physical Asset)	Hardware	• Current status of system in use • Major uses of PC • PC possession status • PC distribution rate
		Software	• S/W status in use
		Network	• Internet use rate • Internet recognition degree • Internet connection method • Utilization field of Internet
		Web Utilization	• Homepage opening rate • Homepage opening language • Homepage operating rate • Homepage reform cycle
	Personnel Component (Intellectual Asset)	Personnel	• Status of informatization education • Status of organization for informatization • Informatization mind of manager • Informatization mind of employees
		Informatization Effect	• Impact of informatization on mould enterprise • Obstacles to the promotion of informatization

2.2 Mould Manufacturing Information System Level

Recent advances in manufacturing and information technologies present promising new strategic alternatives for designing a new manufacturing information system [1, 5]. As the manufacturing environment has recently been marching toward specialization, collaboration, and unification, firms have integrated manufacturing functions and business strategy into their information systems. The total manufacturing information system (TMIS) is a very powerful alternative, which blends recent developments in manufacturing and information technology to achieve a competitive advantage [6].

The overall composition of the TMIS model is shown in Table 2. Also, the TMIS model was compared with the MMIS in the ESMI model.

Various items must be considered when applying the TMIS model to the domestic mould industry. First, marketing area of TMIS is a very important function of the mould industry, but most mould companies are not considering this function at present. In particular, the mould industry need to adopt business systems specialized for the industry, such as an intelligent estimation system, an estimation data management system, and a collaboration system. Second, the ERP System included in engineering area of TMIS should be customized to the mould industry, because the standard ERP

Table 2. Comparison of TMIS with MMIS

TMIS (Total Manufacturing Information System)			MMIS (Mould Manufacturing Information System)	
Category		Information System	Category	Information System
Marketing	Business and Market Analysis	・TQR ・CI ・BIS	MIS	・Business management system ・Purchasing management system ・Cost management system ・Stock/warehouse management system ・Distribution/delivery management system
Engineering	Product Research and Development	Product Development ・LPS, DSS, PDM, ERP	EIS	・CAD/CAE ・PDM ・System for drawing collaboration
		Design & Engineering ・PDM, EMDS, CAD/CAE		
		Procurement ・PDM, ERP, EMDS		
Manufacturing	Computer Integrated Manufacturing	Advanced Manufacturing ・FMS, CAM	PIS	・CAM ・CAPP ・POP ・MES ・Production management system ・Quality control system ・Equipment management system ・Process control system
		Computer Aided Process Planning ・CAPP		
		Automated Material Handling ・AGVS, AS/RS, Robotics		
	Production Planning and Control	・JIT, MRP, MRPII		
	Quality Control	・SPC		
Business	Business Decision Support	・Environmental Analysis Business Strategy Marketing, ・Manufacturing, Quality	-	-

model cannot be applied. The mould industry has a complicated manufacturing process, which does not allow mapping of the complete processing plan and schedule plan. Because customers frequently place rush orders and request mould modification, mould companies need to carry out the BPR (Business Process Reengineering) in advance to introduce the ERP. Third, the domestic mould industry has adopted the small quantity batch production method. And a large portion of the total processing time is taken up by items such as mould processing setup time and mould movement, which engineers must deal with directly. Therefore, it is difficult to introduce an automated material handling system in manufacturing area of TMIS. Finally, although an information system related to business decision support in the business area of TMIS, the introduction of the information system has not been considered due to characteristics of the domestic small and medium-sized mould industry.

In this paper, we analyzed a mould manufacturing information system that can be used from customer's orders to the process of producing final injection mould products (Table 2). The MMIS can be divided into mainly three information systems: MIS, EIS, and PIS. The ERP system was excluded, but detailed modules of ERP were included in the relevant information system. MIS is an information system for business activities related to the mould order receipts. MIS includes many functions, including purchase, cost, stock, and logistics. Meanwhile, EIS, which has played the most important role in the mould production processes, is an information system for various design work necessary for mould production. EIS includes information systems for mould design, mould base design, parts design, and CAE analysis. It also includes the PDM system used to manage the drawings in designs and the collaboration system for real-time drawing collaboration. Finally, PIS is an information system related to mould machining, including CAM work, machining, assembly, and injection mould for NC programming. Through such an analysis, we extracted factors, MIS, EIS, and PIS, from the mould manufacturing information system field and evaluation items about evaluation factors (Table 3).

Table 3. Classification of the Mould Manufacturing Information System

Field	Factors		Items
MMIS (Mould Manufacturing Information System)	MIS	・Business management system ・Purchasing management system ・Cost management system ・Stock/warehouse management system ・Distribution/delivery management system	・Introduction necessity ・Utilization ・Introduction plan within 2 years ・Status of establishment ・Level of practical use
	EIS	・CAD/CAE ・PDM ・System for drawing collaboration	
	PIS	・CAM/CAPP ・POP/MES ・Production management system ・Quality control system ・Equipment management system ・Process control system	

2.3 Collaboration Level

The general concept of collaboration is the process of working together toward a common purpose or goal in which the participants are committed and independent,

with individual and collective accountability for the results of the collaboration, and toward a common benefit to be shared by all participants [7]. However, in the e-business field, the subject of collaboration is not only people but also device and application, and the boundary is continuously expanding [8]. Collaboration between enterprises using information and communication technology has been analyzed from various viewpoints. In the existing studies, however, clear descriptions of the concept of collaboration are very limited.

Collaboration consists of object and time of collaboration. First, the object of collaboration is regarded as enterprise, where collaboration is divided into inbound collaboration and outbound collaboration. Based on the mutual action that an enterprise has in the overall business field, it also could be classified into customer collaboration, supplier collaboration, manufacturing collaboration, and R&D collaboration. Second, according to the time of the collaboration, it could be also divided into asynchronous collaboration and synchronous collaboration [9].

Keskinocak and Tayur classified the form of collaboration into two cases according to the characteristics of the exchanging information [10]. They defined the form of the first case of collaboration as the exchange and sharing of information currently possessed by the enterprises on the supply chain without specific transformation. For an example, we can cite that Wal-Mart share its sales data with its suppliers through a RetailLink System. Thus, with the transparency of the market through information exchange [11], all participants on the supply chain will be able to make a decision by using the shared information, and the entire supply chain can, therefore, change effectively. Secondly, they defined the form of the second case of collaboration as jointly developed information, not the information to be exchanged simply through cooperation between partners on the supply network. In general, this information refers to data related to the development of future products or consumer demands. For instance, we can use analysis data of consumer demands for joint development of new products and forecasting demands.

It was revealed that the individual computer market worked as a vertical collaboration mechanism for price protection [12]. Baumol said that horizontal reform-related collaboration between competing enterprises is helpful to the effectiveness and growth of the overall industry [13]. Likewise, we can divide collaboration into vertical collaboration and horizontal collaboration. Thus, table 4 summarizes the patterns of collaboration, analyzed from existing studies.

If we analyze the characteristics of the domestic mould industry, most mould companies have a pattern of receiving orders from large companies. That is, large companies are their customers. Furthermore, the mould companies that receive orders from these large companies then subcontract set making, machining, injection and measurement to make their moulds. In this supply chain, most mould companies are a subordinate to large companies, and the firms engaged in set making, machining, injection and measurement are also subordinate to the mould companies. Likewise, the pattern of collaboration for mould production takes the form of vertical collaboration. According to the classification criteria for object of collaboration, vertical collaboration in the form of subcontract can be classified as outbound collaboration, customer collaboration, and supplier collaboration. When moulds are manufactured

Table 4. Form of collaboration

Classification	Form		Remark
Object	Enterprise	・Inbound collaboration ・Outbound collaboration	[9]
	Business	・Customer collaboration ・Supplier collaboration ・Manufacturing collaboration ・R&D collaboration	
Time	・Asynchronous collaboration ・Synchronous collaboration		
Information	・Exchange collaboration of information ・Processing collaboration of information		[10]
Relation	・Vertical collaboration		[12]
	・Horizontal collaboration		[13]

within the mould companies, the object of collaboration can be classified as inbound collaboration, manufacturing collaboration, and R&D collaboration. In addition, the Internet-based collaboration system used for drawing working designs can be classified as synchronous collaboration, and the off-line conference for technology meeting, as asynchronous collaboration. As well, according to the nature of information exchanged during collaboration, the collaboration can be classified as simple exchange collaboration and processing collaboration, which can be considered a type of data mining.

In this study, we have extracted evaluation factors by dividing the meaning of collaboration as classified in Table 4 into inter-company collaboration and intra-company collaboration. That is, we regarded vertical collaboration in the supply chain as intra-company collaboration and horizontal collaboration as inter-company collaboration and have extracted an evaluation item for each factor (Table 5).

Table 5. Classification of the Collaboration

Field	Factors	Items
Collaboration	Inter-Collaboration	・Collaboration between departments
	Intra-Collaboration	・Collaboration item ・Field necessary to consult ・Sharable information ・Obstacles to the collaboration ・Collaboration with customers

3 A Case Study and Verification of the Model

3.1 Selection of Weight Factors Using the AHP Technique

In this study, we used the AHP technique, widely recognized for its scientific feasibility to set the relative weight factors of elements that constitute the survey. In order to set the weight factors through the AHP technique, we fixed the weight factors by

using the pair wise comparison method for the developed assessment indicators, and verified by using the Consistency Ratio (CR) to confirm the consistency of responses. We also indicated the weight factors of evaluation areas (fields) and factors by applying the AHP technique to 10 domestic mould experts, results of which are shown in Table 6.

Table 6. Status of Weight Factors

Field	Weight	Factor	Weight
Information/Network Infrastructure	0.2	H/W	0.25
		S/W	0.2
		Network	0.25
		Web Utilization	0.05
		Personnel	0.15
		Informatization effect	0.1
Mould Manufacturing Information System	0.5	MIS(Management Information system)	0.25
		EIS(Engineering Information System)	0.5
		PIS(Production Information System)	0.25
Collaboration	0.3	Collaboration	1

3.2 Survey

Questionnaires were prepared using each items developed from the information/network infrastructure level, mould manufacturing information system level, and collaboration level of the informatization level evaluation framework. The survey was conducted with 152 subjects in total. The objects of the survey were limited to 185 mould companies. The survey response rate was 40.5% (75 subjects/185 subjects), which was considerably high. The survey was conducted for 6 months from September 1, 2005 to February 29, 2006 and was diffused through e-mail and fax. Subsequently, we collected one copy of the questionnaire from each mould company. We employed statistical analysis methods using the statistics package SPSS and Microsoft Excel to analyze the survey responses. Also, reliability analysis, factor analysis, and correlation analysis were used.

3.3 Reliability and Validity

In order to make effective use of this type of questionnaire made through operational definitions, we need to confirm whether they accurately explain the defined concepts. This confirmation is carried out by examining the reliability and validity of the questionnaire. Reliability means the dispersion of measured values appeared when same concept was repeatedly measured. That is, reliability is a concept related to the unsystematic error to be expressed as the stability, consistency, predictability, accuracy and dependability. [14]. There are various methods of reliability analysis, such as the retest method, the multiple forms method, and the internal consistency reliability method. For analyzing reliability, we used the internal consistency reliability method, which is used to measure identical concepts. When using several evaluation areas as in this study, we used Cronbach's Alpha coefficient designed to enhance the reliability of

the measuring tools after finding impeding items of reliability in the tools and excluding them from measuring tools. The value of the Cronbach's Alpha coefficient is generally regarded as reliable if the value is 0.6 or higher [14]. Table 7 indicates the results of reliability in each of the evaluation fields of information/network infrastructure, mould manufacturing information systems, and collaboration. When we analyzed reliability based on the survey results, the Cronbach's Alpha coefficient presented values higher than 0.6 in all areas: the information/network infrastructure (0.668), mould manufacturing information system (0.8553) and collaboration (0.8621). These values indicated no problems in the reliability of the data for each evaluation field.

Table 7. Reliability for the first evaluation framework

Variables	Number of items	Cronbach's Alpha
Information/Network Infrastructure	6	0.668
Mould Manufacturing Information System	3	0.8553
Collaboration	2	0.8621

Next, validity is verified by evaluating how well the measuring items used in the questionnaire explain the contents as defined in this study by mainly through factor analysis. Validity can be divided into content validity, criterion relative validity, construct validity and face validity according to the evaluation method. In this study, we conducted a factor analysis to assess the validity of the questionnaire to evaluate the informatization level. Factor analysis is a method that groups similar variables by using the interdependence among variables.

Table 8 shows the result of the factor analysis for measuring items in this study. We conducted the evaluation by including informatization effect and personnel items

Table 8. Rotated Component Matrix (a)

	Component		
	1	2	3
MIS	.875		
PIS	.869		
EIS	.792		
Informatization effect	.493		
Personnel	.472		
H/W		.832	
Web utilization		.762	
Network		.674	
S/W		.576	
Intra-Collaboration			.964
Inter-Collaboration			.957

into the information/network infrastructure field. As the result of factor analysis, however, the two items were found to belong to the mould manufacturing information system field. Also, the Cronbach's Alpha coefficient remained at below 0.6, indicating problems in the reliability of data in each evaluation field. To reflect these results, we conducted the factor analysis once again by excluding the informatization effect and personnel items from the evaluation framework (Table 9).

Cumulative loading value of the existing factors was 67.097, making it possible to explain about 67% of the total, while cumulative loading value in the analysis of validity excluding the two items was 74.089, explaining about 74% of the total. This analysis result can be judged as having greater explanatory capacity than the model prior to revision. We conducted the analysis of reliability once again on the basis of the results of Table 9 (Table 10). The Cronbach's Alpha coefficient was higher than 0.6, proving sufficient reliability.

Table 9. Rotated Component Matrix (a)

	Component		
	1	2	3
MIS	.883		
PIS	.874		
EIS	.809		
H/W		.857	
Web utilization		.754	
Network		.682	
S/W		.608	
Intra-Collaboration			.968
Inter-Collaboration			.962

Table 10. Reliability for the second evaluation framework

Variables	Number of items	Cronbach's Alpha
Information/Network Infrastructure	4	0.6041
Mould Manufacturing Information System	3	0.8553
Collaboration	2	0.8621

Figure 3 indicates the evaluation framework from the information view of the ESMI model. The items to evaluate the informatization level were information/network infrastructure, mould manufacturing information systems, and collaboration, with personnel and information effect items as potential domains. In the evaluation field of Figure 2, we classified personnel and informatization effect factors as information/network infrastructure field, but realized that the evaluation framework is a more effective if two items were classified as another field through the verification of reliability and validity.

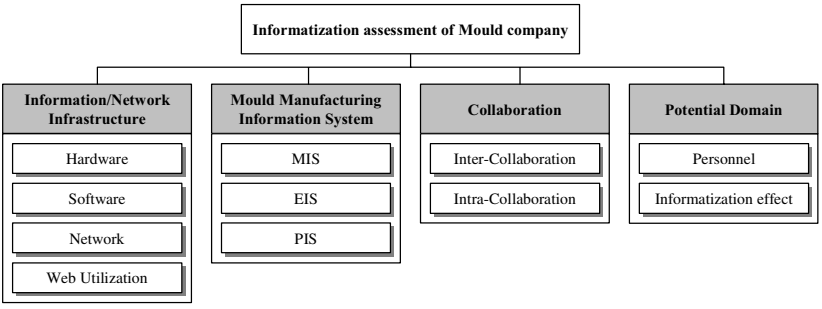


Fig. 3. Evaluation framework of the ESMI model (2nd)

4 Conclusions

In this study, we propose an ESMI model that can evaluate the informatization level of the mould companies. The ESMI is a model that integrates the information and business viewpoints of the informatization level. Based on this model, an evaluation framework from the business viewpoint was presented. Evaluation fields were information/network infrastructure, mould manufacturing information systems, and collaboration. Particularly, the classification of the MMIS (Mould Manufacturing Information System) will enable the mould companies to introduce an effective information system. We classified factors in each field and composed evaluation items per factor. Next, the model was verified and framework was formulated. A questionnaire based on the AHP method was used to conduct a survey of the evaluation items. Seventy-five survey responses analyzed to ensure the reliability and validity of the questionnaire items. Ultimately we verified the ESMI model from verification process.

The evaluation model developed in this study can be used to periodically diagnose the informatization level of mould companies as well as to measure the ROI (Return of Investment) in informatization. That is, the necessity of an information system can be evaluated before introducing the information system and the effects after introducing it can be measured.

If the mould companies introduce informatization after evaluating their informatization levels, they can promote balanced and systematic informatization and use the evaluation results effectively as basic data and guidelines to promote continuous informatization. Also, when the mould industry introduces various information systems using up-to-date Internet technology, such as e-collaboration and e-commerce, they can apply the evaluation model of informatization level offered by this study.

References

1. Coates, J.: Manufacturing in the 21st century. International Journal of Manufacturing Technology and Management 1(1), 42–59 (2000)
2. Teo, T.S.H., King, W.R.: Integration Between Business Planning and Information Systems Planning: An Evolutionary- Contingency Perspective. Journal of Management Information Systems 14(1), 185–214 (1997)

3. Lederer, A.L., Sethi, V.: Key Prescriptions for Strategic Information Systems Planning. Journal of Management Information Systems 13(1), 35–62 (1996)
4. Davenport, T.H., Linder, J.: Information management infrastructure: the new competitive weapon. In: Proceedings of the 27th Hawaii International Conference on System Sciences, pp. 885–896 (1994)
5. Tan, D.S., Uijttenbrock, A.A.: Information infrastructure management: a new role for IS managers. Information System Management 14(4), 33–41 (1997)
6. Lee, C.Y.: Total manufacturing information system: a conceptual model of a strategic tool for competitive advantage. Integrated Manufacturing Systems 14(2), 114–122 (2003)
7. Light, M., Bell, M., Halpern, M.: What is Collaboration? Virtual Team Success Factors. Gartner Research Note COM-14-4302 (2001)
8. Arevolo, W.: Rethinking Collaboration, Business Challenges and Opportunities. Gartner Research Note COM-12-8881 (2001)
9. Hayward, S.: Collaboration: From Problem to Profit. Gartner Research Note COM-12-7261 (2001)
10. Keskinocak, P., Tayur, S.: Quantitative analysis for Internet-enabled supply chains. INTERFACES 31(2), 70–89 (2001)
11. Phillips, C., Meeker, M.: The B2B internet report - Collaborative commerce. Morgan Stanley Dean Witter (2000), http://www.morganstanley.com/institutional/techresearch/pdfs/ b2bp1a.pdf
12. Lee, H.L., Padmanbhan, V., Taylor, T.A., Whang, S.: Proce protection in the personal computer industry. Manegement Science 46, 467–482 (2000)
13. Baumol, W.J.: When is inter-firm coordination beneficial? The case of innovation. International Journal of Industrial Organization 19, 727–737 (2001)
14. Nunnally, J.C.: Psychometric theory, 2nd edn. McGraw-Hill, New York (1978)

Framework of a Collaboration-Based Engineering Service System for Mould Industry

Chang Ho Lee[1] and Yong Ju Cho[2]

[1] Department of Industrial Sysem Engineering, Yonsei University,
134, Seodaemun-gu, Seoul, 120-749, Korea
changho.lee@yonsei.ac.kr
[2] i-Manufacturing Center, Korea Institute of Industrial Technology,
7-47, Songdo-dong, Yeonsu-gu, Incheon Metropolitan City 406-840, Korea
yjcho@kitech.re.kr

Abstract. Currently, the environment of manufacturing industry is becoming a distribution, specialization and collaboration. Particularly, the mould industry, the basis of the Korea manufacturing industry, is also experiencing those changes and the requirements of customers for mould companies are becoming increasingly diverse. To cope this, mould enterprise is endeavoring for satisfaction of the requirements in order processing, mould & part design, mould machining, and pre-injection which are unique jobs of mould enterprise. But, a lot of inferiority is occurring by a poor production environment. In order to troubleshoot this inferiority, the engineering services such as CAE analysis and inspection are absolutely required. The function of engineering service provides major businesses within mould company with new needs and rules. In this paper, the framework of e-engineering service system is implemented with the e-business and the e-commerce form. Also, functions in framework of e-engineering service system are implemented by using agent modules.

Keywords: Collaboration, Mould industry, Internet & Information Technology, Engineering Service.

1 Introduction

1.1 Changes in the Manufacturing Environment

To provide promptly niche products and services to the consumer, SMEs must take part in supply chain corresponding to a manufacturing industry. On the other hand, the role of SMEs in manufacturing supply chains is increasing due to several effects, such as:

- Larger companies desire to focus on their core competencies. As a result, many are out-sourcing activities not among the core.
- Information technology has shifted the equilibrium between co-ordination costs and transaction costs in favor of SME participation.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 33 – 44, 2007.
© Springer-Verlag Berlin Heidelberg 2007

- The larger players in many sectors have come together to create electronic marketplaces. For example, several of the large automobile firms have co-operated to set up purchasing networks on-line.

Also, from the mid-1990s there have been two trends which can work toward strengthening the position of SMEs within supply chains:

- Concentration upon core competencies,
- Tighter cooperation among the firms within a supply chain.

The above suggests that, in the current manufacturing industries, crucial issues, such as collaboration (cooperation), concentration and integration are emerging.

1.2 SMEs and Internet and Information Technology

Internet and information technologies are evolutionary developments, relatively inexpensive, and public networks [1]. Viewed from a manufacturer's perspective, Internet and information technology can be seen to provide opportunities to communicate and interact with suppliers and more significantly, customers/consumers, in ways not previously possible with EDI technology [2]. Also, the adoption of internet and information technology amongst SMEs appears to be driven by external factors such as more technologically advanced and commercially powerful customers [3].

These internet and information technologies can support better cooperation. With better information, producers can better provide what the customer desires, at lower cost both for the buyer and seller.

The interactions supported by the Internet span a range of integration. This range can be split into:

- Informational cooperation,
- Operational cooperation.

Informational cooperation can involve SMEs providing prices, availability, or order status through the Internet. Operational cooperation is deeper and requires tighter integration. Here, the SMEs not only share data but co-ordinate activities, such as inventory allotment or forecasting.

Especially among the manufacturing industries, automotive industries, such as GM and Ford, have introduced virtual market in which internet and information technologies are utilized [4]. These virtual markets will allow the firms to buy parts and components from thousands of suppliers and can be created by a large firm within the supply chain (such as those created by GM and Ford) or by a third party (such as i2i.com). The following are virtual markets currently running against manufacturing industries.

- http://www.hub-m.com/default.asp
- http://www.nc-net.or.jp
- http://www.ntma.org

1.3 The Necessity of Knowledge Collaboration in Mould Industry

As aforementioned, B2B, e-commerce and Internet web site related to manufacturing industries are being widely spread to various manufacturing industries. Especially, the

mould industry, which is the basis of manufacturing industries and is a kind of ETO industry, needs a new paradigm to cope with such changes properly and to enhance competitiveness.

Companies making moulds do not utilize mass-production facilities but facilities which can produce small quantity, various types of moulds. Because of this aspect, they should have the ability to meet customer's demand faster and more dynamically and to cope with the changes of environments in and outside of the companies. Such mould industries need to shift to a new system in which Internet and information technologies are employed, due to the following causes. First, instead of developing new products by its own company, it is important to reduce production time by making a product based on customer's request. Second, most mould companies are medium size, and 81.5% of 2,500 medium size companies have employees of less than 20 people. Due to this aspect, they lack investment power, man power, and information, which are the basis of R&D. Third, lack of professionals in the area of CAE/CAD/CAM, and system programming, results in the reduction of production rate and the poor quality of moulds. In addition, the survey conducted on questions related to information technology for mould companies [5], shows that demand is highest concerning sharing knowledge associated with design (Fig. 1 and Fig. 2).

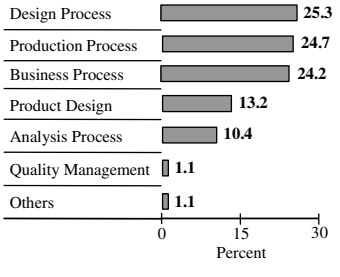


Fig. 1. Collaboration items

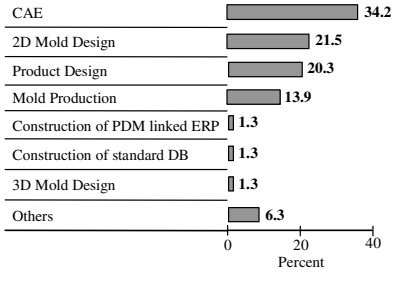


Fig. 2. Consulting fields

A shift in the industrial usage of IT systems has been identified as moving from the application of "data-driven environments" to that of "co-operative information and knowledge-driven environments" [6, 7]. Consequently, to vitalize mould industries, it is necessary to develop a system in which Internet and information technologies are combined, resulting in knowledge related to design being shared with others working on mould design.

1.4 The Purpose and Organization of This Study

In this study, cooperation of work and the sharing of knowledge associated with design are defined as cooperation of work and sharing of knowledge in engineering service, with which e-engineering service is a combination of Internet and information technology. E-commerce encompasses a concept of CALS (Commerce At Light Speed) which reduce raw-material cost and lead time for the development of new

products by sharing information, such as design/development of products, manufacturing, transportation, and disposal, with few industries related to design. Thus, e-engineering service can be considered as e-commerce.

Medium size industries, including manufacturing companies which produces parts by customer's order, have relatively little capital, manpower, and technology. In order for them to cope with the rapid spreading of e-commerce with competitiveness, a new agent technique, as a new paradigm, by which small quantity, various kinds of parts can be made within a short time, is emerging [8]. In this study, a framework to establish e-engineering service combined with such agent technology is proposed.

This study consists of the following. In chapter 2, importance of engineering service in mould industries is addressed through IDEF0 modeling. In chapter 3, business models of e-engineering service to be embodied in this study are analyzed. In chapter 4, the structure of cooperative agents is proposed, and their function defined. In addition, overall framework of the agents is proposed. In chapter 5, the total process, in which e-engineering service is performed, is explained. In the last part, conclusions and future research are addressed.

2 Significance of e-Engineering Service in Mould Industry

Figure 3 shows kinds of engineering service performed off-line in mould industries and overall concept for implementation of service to be proposed in this study. It is shown that a virtual market is connected to the Internet through which customers request services and the services are provided by service providers. The service providers are composed of external institutes specialized in corresponding services and specialists, and located in various places. Customer can request services from industries available in the supply chain of mould industries. Likewise, customers are located in various areas. Currently mould industries and manufacturing companies receive orders of various parts from customers and manufacture them. In this stage, to satisfy the need of the customers, collaboration is necessary for the design and manufacturing process of parts, which require 3D modeling and use of CAE and CAM. In the previous off-line service system, market competitiveness is weak due to delay in time and economical waste. Thus, it is imperative to establish an e-engineering system by which engineering service and data can be obtained on the real time basis.

From the viewpoint of providers who take part in e-engineering service, they have the following advantages. First, engineering service served through the Internet enables them to identify the tendencies of requests from clients, and the information can be transferred to an knowledge system easily. Second, prompt services to the clients are available. Third, due to the fact that the service is not influenced by time and space, many specialists can be involved in the service. Similarly, the clients have the following advantages. First, time and effort to receive engineering service can be reduced, due to the avoidance of a client's trip. Second, the content of services can be reviewed carefully, and accurate and effective service can be received. Third, clients can receive services at a low price due to many service providers available.

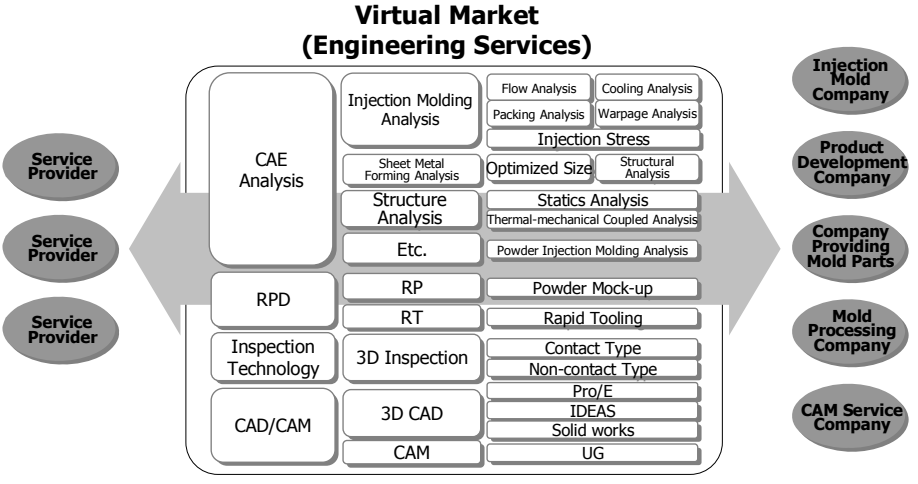


Fig. 3. Diagram of the concept of executing engineering service

2.1 AS-IS and TO-BE Process of Mould Industry Using IDEF0

The IDEF suite of enterprise modeling approached, which comprises IDEF0, IDEF1, IDEF1x, IDEF3 and other graphically based modeling notations have been applied extensively in support of large industrial engineering projects [9, 10]. IDEF0 was

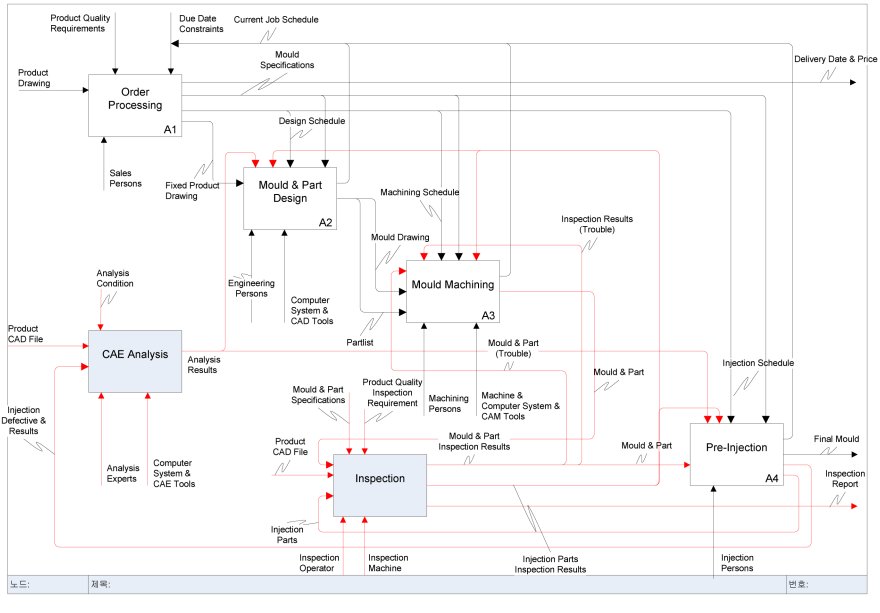


Fig. 4. AS-IS and TO-BE process of mould company

developed in order to represent activities or processes (comprising partially ordered sets of activities) that are carried out in an organized and standard manner [11]. In this study, the result of a modeling on AS-IS process to which engineering service is not applied and TO-BE process to which engineering service is applied, is shown in Figure 4. The modeling was performed using IDEF0.

Major work executed in mould industries can be classified into order processing, mould & parts design, mould machining, and pre-injection. The AS-IS process in which existing work does not reflect engineering services, including CAE analysis and inspection, is considered to be simple. In the case where defects in moulds are found and there are problems with the existing work, all processes should be repeated again. However, there are not suitable solutions for them due to the absence of absolute cause analyses. On the contrary, the TO-BE process, in which CAE analysis and inspection are applied, is relatively complicated. The IDEF0 modeling result shows that the output of previous activities is related to the input of engineering service activities and the output of engineering service activities is related to the input and control item of previous activities.

3 Analysis of Type About e-Engineering Service Trade Method

E-commerce is defined as "all business performed online" in a broad meaning [12], and its meaning includes transferring invisible services, such as physical goods and information, online. Therefore, all transactions associated with online marketing, ordering, payment, and delivery, include legal services supplied online and joint research conducted online, and joint design. E-engineering proposed in this study can be taken into account as e-commerce. Thus, in this study, the e-engineering service model is analyzed in terms of a business model. At first, to analyze the aspect of e-engineering service transactions, an e-engineering service model which can be used for classifying existing e-commerce based on five items, is proposed (Table 1). First, in the item of "business participants", engineering service can be considered as B2B. Second, in the item of "source of revenue", a client pays the service fee after he receives engineering service. Third, n number of service providers and n number of clients perform engineering services on a virtual market. Fourth, engineering service

Table 1. Analysis of the trade aspect of e-engineering service

Items	Category					
Business participants	B2C		**B2B**		C2C	
Source of revenue	Advertisement		Brokerage	**Toll**		Membership Fee
Interaction	1 to 1		1 to N		**N to N**	
Business method	Retail	Auction	Reverse Auction	Portal	Catalog	**Sale to order**
Business model	Internet business model			**Off-line business model transplanted in Internet business model**		

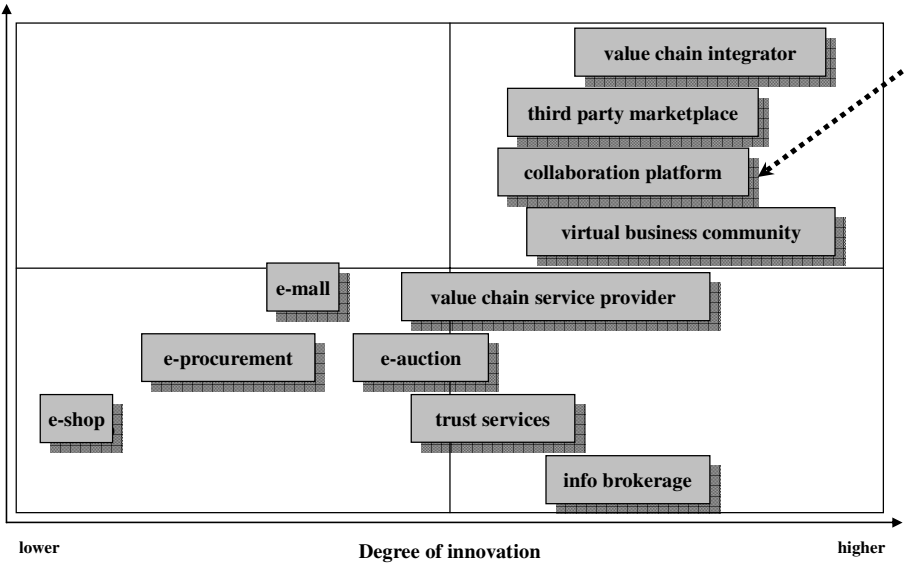


Fig. 5. Business models – classification

is a kind of business performed based on a customer's order and sale, that is, a service is carried out after a client request services and then a specialist performs the services. Fifth, engineering service covered in this study performed online takes on a tendency of a business model that performed off-line.

Timmers proposed eleven more detailed business models (Figure 5) using those shown in Table 1 [13]. Of the models proposed, the engineering service model is very similar to the collaboration platform model from the following definition: collaboration platform provides a set of tools and an information environment for collaboration between enterprises. This can focus on specific functions, such as collaborative design and engineering, or in providing project support with a virtual team of consultants [13]. As a result, unlike e-shop, e-procurement, and e-auction which perform simple transaction of products and works associated with the buying and selling of products, the e-engineering service can be a more innovative business model.

4 Collaborative Agents for e-Engineering Service

Recently, the manufacturing environment changed from a static to a dynamic environment. To cope with this trend, agent technique emerges as a new paradigm in manufacturing systems. This means that as the demands of clients are diverse, manufacturing environments need to be flexible to meet their demands. In addition, diversification of the manufacturing environment and an increase in transactions require agent technique to support automation of the work.

As aforementioned, the e-engkineering model proposed in this study which is used to buy and sell knowledge online can be considered e-business or e-commerce. That is, agent techniques which are used in e-commerce can be applied to e-engineering

service model. Likewise, there are many areas where agent techniques can be applied to [14]. The previous trait of the agents needs to be applied as it is. The content of agent properties is as follows. First, autonomy means that agents operate without direct intervention of humans or other agents have control over their actions and internal state [15]. Second, pro-activeness means that agents do not simply act in response to their environment, but they are able to exhibit goal-oriented behavior by taking the initiative. Third, reactivity means that agents perceive their environment, including the physical world, users, and other agents, and respond in timely fashion to changes observed. Fourth, sociability means that agents interact with other agents via some kind of communication language. Fifth, mobility indicates that agents might move from one machine to another. Sixth, learning provides agents with the ability to adopt their behavior on the basis of their experience [16, 17].

There are a number of different kinds of intelligent agents that work together autonomously and co-operate with each other to perform different tasks for the engineering service. Figure 6 shows the agent hierarchy. A detail of each agent is as follows.

First, MPA (Monitoring & Pre-evaluation Agent) is activated when a client requests an engineering service. Its major function includes a request KMA client's information search, service result search, and specialist search from the knowledge database. If there is a service request which is similar to the previous service, corresponding service result is sent to the client. This function is useful since most of mould industries manufacture similar parts. For instance, in the case of TVs, only their sizes are different. Thus, a database of service results needs to be established, and an expert system can be utilized to check similar results from the database. In the next stage, the content of the service is sent to a specialist and service cost and service reporting date are sent to a client. In the last stage, once implementation of the service is decided, CPA (Collaborative Process Agent) is initialized.

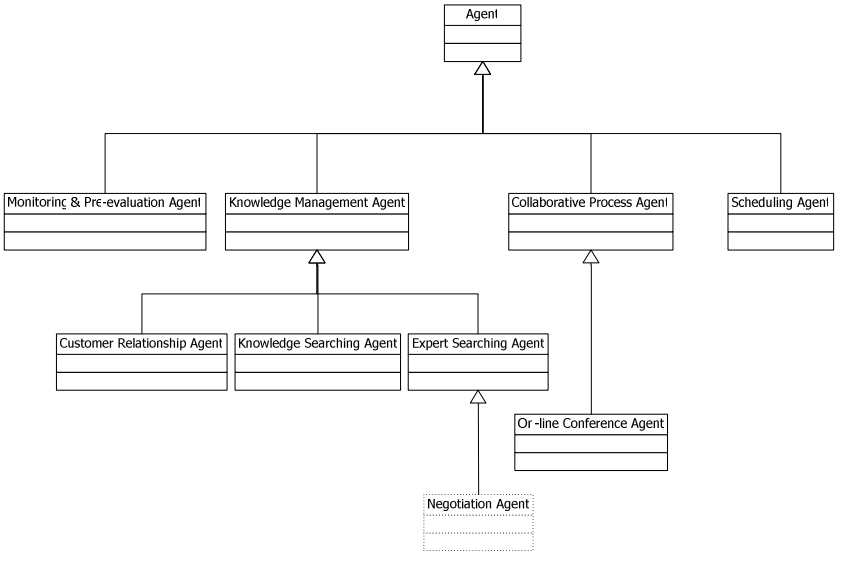


Fig. 6. The agent hierarchy

Second, KMA (Knowledge Management Agent) assists Information retrieval, information filtering, or other information manipulation. It has three sub-class agents, CRA (Customer Relationship Agent), KSA (Knowledge Searching Agent), and ESA (Expert Searching Agent). CRA reasons about capabilities of, or relationship between, clients' resources. This agent is responsible for keeping track of the changes of CRM (Customer Relationship Management) in dynamic environment. And KSA searches and gathers the relevant service result and information from internal resources, such as database and file systems. Also, checked results need to be sent to MPA. ESA informs a registered specialist of requested service. The due date and the cost can be negotiable to perform the service at the best possible conditions. Figure 6 shows the dotted box which indicates the content of negotiation. The last function is to select the most suitable specialist and inform MPA of him.

Third, CPA (Collaborative Process Agent) initiates SA and OCA. Only this agent performs the function of a broker.

Fourth, SA (Scheduling Agent) reports to the client the status of the service being performed.

Fifth, OCA (On-line Conference Agent) supports the work which makes it possible for services to be performed online. Its major function includes collecting additional references required for service from the client and KSA. When the conference between the client and the service provider is needed, cooperative work is possible using engineering service system.

Figure 7 shows the e-engineering service model displaying the overall framework.

5 An Example

In order to illustrate our concepts and techniques, we will demonstrate a workflow taking place in an e-engineering service system, with agent collaborative support (Fig. 7).

A Client makes a service implementation plan and requests an engineering service (step1), after which MPA is initialized (step 2). MPA provides CRA with client's information (step 3a), and CRA checks its database and provides the client with the service (step 3b). In the next stage, MPA requests KSA to check a similar service based on the content requested by the client, and KSA uses expert system to check a service result database and to provide the best results (step 4a). KSA provides the client with the results checked (step 4b). After that, MPA requests ESA a specialist from an expert database, ESA checks the request from the client and selects a specialist using a registered expert database, and the result is sent (step 5). MPA informs a selected service provider of the requested service, and a service cost, calculated based on the content of service and service reporting date needs to be sent to the client (step6). MPA initializes CPA (step 7). CPA initializes SA (step 8), and an activated SA sends the extent of the progress to the client (step 9a & 9b). CPA initializes OCA and the activated OCS collects additional references and data through the interaction between the client and the service provider (step 11a & 11b). In the case where the client needs to be in cooperation with the service provider online to perform the service, engineering service system needs to be called (step 11c). The function of engineering service system includes discussion with viewing a mould drawing in a virtual

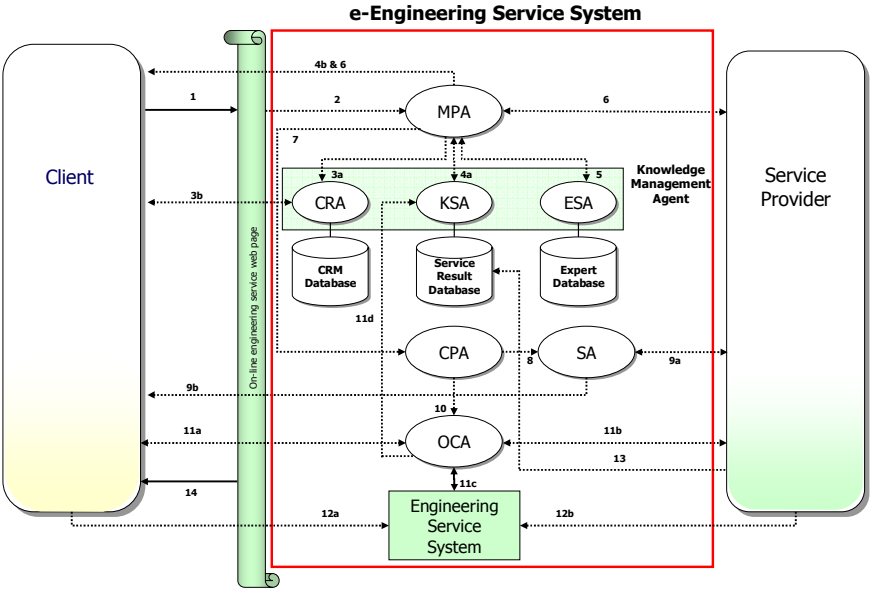


Fig. 7. Framework of e-engineering service system and service workflow

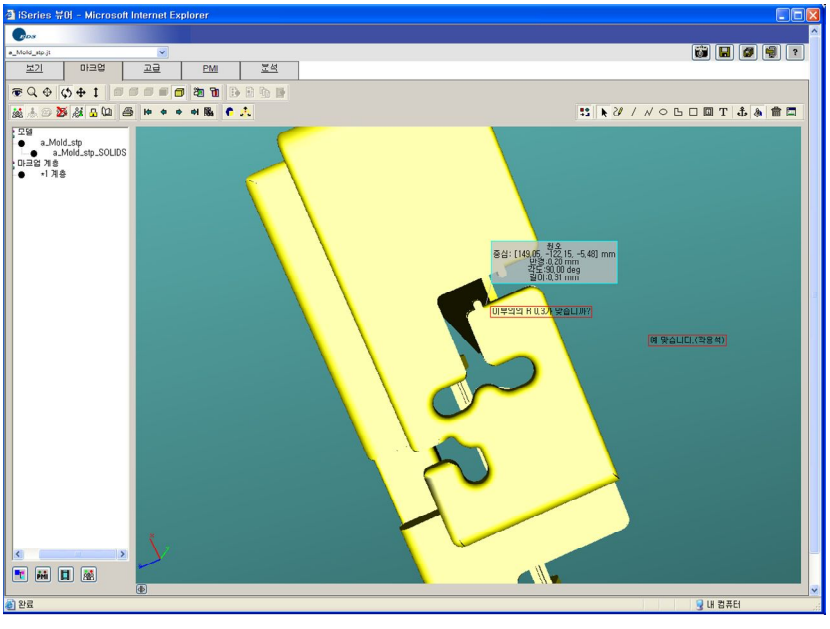


Fig. 8. Snap-shot of on-line conference

space (step 12a & 12b). The service provider saves service results in the database of service result (step 13). The final stage is that the client confirms the result on the web page (step 14).

Now a project to establish cooperative work in the area of design in conventional manufacturing industries is in progress in the name of the project entitled e-manufacturing, which enables rapid supply of products at global manufacturing environment (http://www.i-mftg.com/default.aspx). The major content of this business is as follows.

- Establishment of the system for cooperative work in the area of design to develop products,
- Cooperative service among industries, parts, and development of products,
- Establishment of engineering service on difficult technologies in medium size industries.

The framework of e-engineering service system, proposed in this study, is being applied and developed. Figure 8 illustrates a snap-shop displaying cooperative work being performed online in the collaborative hub system (step 12a & step 12b).

6 Conclusions and Future Works

This study focused on supporting design work which is the core of mould industries, which is the basis of manufacturing industries. The main content includes design of framework for the development of an online system which enables transaction of engineering knowledge.

The following is the content addressed in this study. First, AS-IS process of work in mould industries through IDEF0 modeling and TO-BE process which includes engineering work were analyzed. Second, e-business or e-commerce concept was applied to e-engineering service model. Third, to make accurate analysis of e-engineering service model, the model was analyzed in terms of business model. Fourth, agent concept was applied to e-engineering service model. Fifth, the agent hierarchy of e-engineering service model was designed, and all the framework of e-engineering service model, which is the basis of agent, was also designed. In addition, the example of a workflow, by which the service is performed, was proposed, and the snap shot, which is being developed, is proposed.

The future work includes the development of an agent module proposed in this study and its integration. Knowledge accumulation is required to enhance the use of knowledge management agent proposed in this study. That is, it is necessary to develop a method to accumulate data on service result, client, and service provider and to utilize the knowledge. For instance, methods, including QFD (Quality Function Deployment), nerve system, and CRM technique, need to be applied. Finally, the developed system must be expanded to all types of manufacture.

References

1. Beach, R.: Adopting Internet technology in manufacturing: a strategic perspective. Production Planning & Control 15(1), 80–89 (1992)
2. Jelassi, T., Leenen, S.: An e-commerce sales model for manufacturing companies: a conceptual framework and a European example. European Management Journal 21(1), 38–47 (2003)

3. Quayle, M.: E-commerce: the challenge for UK SMEs in the twenty-first century. International Journal of Operations & Production Management 22(10), 1148–1161 (2001)
4. Stedman, C.: Race heats up for e-supply chains. Computerworld 33(45), 16–18 (1999)
5. Cho, Y.J., Leem, C.S., Shin G.T.: An assessment of the level of informatization in the Korea mold industry as a prerequisite for e-collaboration: an exploratory empirical investigation. International Journal of Advanced Manufacturing Technology (Accepted) (2005)
6. Scheer, A.W., Kruse, C.: ARIS-framework and toolset. A comprehensive business process reengineering methodology. In: Proceedings of the 4th International Conference on Automation, Robotics & Computer Vision (ICARCA), Singapore, pp. 327–331 (1994)
7. Vernadat, F.B.: Enterprise Modelling and Integration: Principles and Applications. Chapman & Hall, London (1996)
8. Shu, S., Norrie, D.H.: Patterns for Adaptive Multi-Agent Systems in Intelligent Manufacturing. In: Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems, Leuven, Belgium, pp. 67–74 (1999)
9. Ang, C.L., Khoo, L.P., Gay, R.K.L.: A comprehensive modelling methodology for development of manufacturing enterprise system. International Journal of Production Research 37(17), 3839–3858 (1999)
10. Mayer, R.J., Menzel, C.P., Painter, M.K., deWitte, P.D., Blinn, T., Perakath, B.: Information Integration for Concurrent Engineering IDEF3 Process Description Capture Method Report. Knowledge Based Systems Inc. (1995)
11. Kim, C.H., Weston, R.H., Hodgson, A., Lee, K.H.: The complementary use of IDEF and UML modelling approaches. Computers in Industry 50, 35–56 (2003)
12. European Commission: European Initiative in Electronic Commerce. COM(97) 157, April 1997, Ch. 1 Also available at http://www.cordis.lu/esprit/src/ecomcom.htm
13. Timmers, P.: Business Models for electronic Markets. Electronic Markets 8(2) (1998)
14. Papazoglou, M.P.: Agent-oriented technology in support of e-business. Communications of the ACM 44(4), 71–77 (2001)
15. Verharen, E.: A Language-Action perspective on the design of cooperative information agents. Ph.D. thesis, Katholieke Universiteit Brabant, Tilburg (1997)
16. Wooldridge, M.: Intelligent Agents. In: Weiss, G. (ed.) Multiagent systems: A modern approach to distributed artificial intelligence, pp. 27–78. MIT Press, Cambridge (1999)
17. Dignum, F., Cortés, U. (eds.): Agent-Mediated Electronic Commerce III. LNCS (LNAI), vol. 2003. Springer, Heidelberg (2001)

Meta-modelling Syntax and Semantics of Structural Concepts for Open Networked Enterprises

Mohamed Bouhdadi, Youssef Balouki, and El maati Chabbar

Mohammed V University, Department of Mathematics & Computer Science,
Rabat Morocco
{bouhdadi, balouki, chabbar}@fsr.ac.ma

Abstract. The Reference Model for Open Distributed Processing (RM-ODP) defines a framework within which support of distribution, interoperability and portability can be integrated. However other ODP standards have to be defined. We treat in this paper the need of formal notation for structural concepts in the enterprise language. Indeed, the ODP viewpoint languages are abstract in the sense that they define what concepts should be supported not how these concepts should be represented. One approach to define formal semantics of a language is denotational elaborating the instance denoted by a sentence of the language in a particular context. Using the denotational semantics in the context of UML/OCL meta-modelling approach and the denotational semantics we define in this paper syntax and semantics for a fragment of ODP structural concepts defined in the RM-ODP foundations part and in the enterprise language. These specification concepts are suitable for describing and constraining ODP enterprise viewpoint specifications.

Keywords: RM-ODP, Enterprise Language, Structural Concepts, Denotational Semantics, UML/OCL.

1 Introduction

The rapid growth of distributed processing has led to a need for coordinating framework for the standardization of Open Distributed Processing (ODP). The Reference Model for Open Distributed Processing (RM-ODP) [1-4] provides a framework within which support of distribution, networking and portability can be integrated. It consists of four parts. The foundations part [2] contains the definition of the concepts and analytical framework for normalized description of (arbitrary) distributed processing systems. These concepts are grouped in several categories. The architecture part [3] contains the specifications of the required characteristics that qualify distributed processing as open. It defines a framework comprising five viewpoints, five viewpoint languages, ODP functions and ODP transparencies. The five viewpoints are enterprise, information, computational, engineering and technology. Each viewpoint language defines concepts and rules for specifying ODP systems from the corresponding viewpoint. The enterprise viewpoint is the first specification of an open distributed system. It is concerned with the purpose, scope and policies for the ODP system.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 45 – 54, 2007.
© Springer-Verlag Berlin Heidelberg 2007

However, RM-ODP can not be directly applicable [3]. In fact RM-ODP only provides a framework for the definition of new ODP standards. These standards include standards for ODP functions [6-7]; standards for modelling and specifying ODP systems; standards for programming, implementing, and testing ODP systems. Also RM-ODP recommends defining the ODP types for ODP systems [8].

In this paper we treat the need of formal notation of ODP viewpoint languages. Indeed, the viewpoint languages are abstract in the sense that they define what concepts should be supported, not how these concepts should be represented. The RM-ODP uses the term language in its broadest sense:" a set of terms and rules for the construction of statements from the terms;" it does not propose any notation for supporting the viewpoint languages. A formal definition of the ODP viewpoint languages would permit to test the conformity of different viewpoint specification and to verify and validate each viewpoint specification. In the current context of software engineering methods and formal methods, we use the UML/OCL denotational meta-modelling, semantics to define semantics for structural specification concepts in ODP enterprise language. The part of RM-ODP considered is a subset for describing and constraining the structure of ODP enterprise viewpoint specifications. It consists of modelling and specifying concepts defined in the RM-ODP foundations part and concepts in the enterprise language. The syntax domain and the semantics domain are defined using UML and OCL. The association between the syntax domain and the semantics domain are defined in the same models. The semantics of UML model is given by constraining the relationship between a model and possible instances of that model. That is, constraining the relationship between expressions of the UML/OCL abstract syntax for models and expressions of the UML/OCL abstract syntax for instances. This is done using OCL.

The paper is organized as follows. Section 2 describes related works. Section 3 we introduce the subset of concepts considered in this paper namely the object model and main structural concepts in the enterprise language. Section 4 defines the meta-model for the language of the model: object template, interface template, action template, type, and role. The meta-model syntax for considered concepts consists of class diagrams and OCL constraints. Section 5 describes the UML/OCL meta-model for instances of models Section 6 makes the connection between models and their instances using OCL. A conclusion end perspectives end the paper.

2 Related Work

The languages Z [9], SDL [10], LOTOS [11] and, Esterelle [12] are used in RM-ODP architectural semantics part [4] for the specification of ODP concepts. However, no formal method is likely to be suitable for specifying every aspect of an ODP system. In fact, these methods have been developed for hardware design and protocol engineering. The inherent characteristics of ODP systems imply the need to integrate different specification languages and different verification methods.

Elsewhere, this challenge in formal methods world is the same in software methods. Indeed, there had been an amount of research for applying the Unified Modelling Languages UML [13] as a notation for the definition of the syntax of UML itself [14-16]. This is defined in terms of three views: the abstract syntax, well-formedness

rules, and modeling elements semantics. The abstract syntax is expressed using a subset of UML static modelling notations that is class diagrams. The well-formedness rules are expressed in Object Constrains Language OCL [17]. OCL is used for expressing constraints on object structure which cannot be expressed by class diagrams alone. A part of UML meta-model itself has a precise semantics [18, 19] defined using denotational meta-modelling semantics approach. A denotational approach [20] is realized by a definition of the form of an instance of every language element and a set of rules which determine which instances are and are not denoted by a particular language element. The three main steps of the approach are : (1) define the meta-model for the language for models (syntax domain), (2) define the meta-model for the language of instances (semantics domain), and (3) define the mapping between these two languages

Furthermore, for testing ODP systems [2-3], the current testing techniques [21], [22] are not widely accepted and especially for the enterprise viewpoint specifications. A new approach for testing, namely agile programming [23], [24] or test first approach [25] is being increasingly adopted. The principle is the integration of the system model and the testing model using UML meta-modelling approach [26, 27]. This approach is based on the executable UML [28]. In this context OCL is used to specify the invariants [19] and the properties to be tested [24]. The OCL invariants are defined using the UML denotational semantics and OCL itself.

In this context we used the meta-modelling syntax and semantics approach in the context of ODP languages. We defined syntax of a sub-language for the ODP QoS-aware enterprise viewpoint specifications [29]. We also defined a UML/OCL meta-model semantics for structural concepts in ODP computational language [30].

3 The RM-ODP

RM-ODP is a framework for the construction of open distributed systems. It defines a generic object model in the foundations part and an architecture which contains the specifications of the required characteristics that qualify distributed processing as open. We overview in this section the core structural concepts for ODP enterprise language. These concepts are sufficient to demonstrate the general principle of denotational semantics in the context of ODP viewpoint languages.

3.1 The RM-ODP Foundations Part

The RM-ODP object model [3] corresponds closely to the use of the term data-model in the relational data model. To avoid misunderstandings, the RM-ODP defines each of the concepts commonly encountered in object oriented models. It underlines a basic object model which is unified in the sense that it has successfully to serve each of the five ODP viewpoints. It defines the basic concepts concerned with existence and activity: the expression of what exists, where it is and what it does. The core concepts defined in the object model are object and action.

An object is the unit of encapsulation: a model of an entity. It is characterized by its behavior and, dually, by its states.

Encapsulation means that changes in an object state can occur only as a result of internal actions or interactions.

An action is a concept for modeling something which happens. ODP actions may have duration and may overlap in time. All actions are associated with at least one object: internal actions are associated with a single object; interactions are actions associated with several objects.

Objects have an identity, which means that each object is distinct from any other object. Object identity implies that there exists a reliable way to refer to objects in a model. When the emphasis is placed on behavior an object is informally said to perform functions and offer services, theses functions are specified in terms of interfaces. It interacts with its environment at its interaction points which are its interfaces. An interface is a subset of the interactions in which an object can participate. An ODP object can have multiple interfaces. Like objects, interfaces can be instantiated.

The other concepts defined in the object model are derived from the concepts of object and action; those are class, template, type, subtype/supertype, subclass/ superclass, composition, and behavioral compatibility.

Composition of objects is a combination of two or more objects yielding a new object.

An object is behaviorally compatible with a second object with respect to a set of criteria if the first object can replace the second object without the environment being able to notice the difference in the object behavior on the basis of the set of criteria.

A type (of an $<x>) is a predicate characterizing a collection of <x>s. Objects and interfaces can be typed with any predicate. The ODP notion of type is much more general than of most object models. Also ODP allows ODP to have several types, and to dynamically change types.

A class (of an <x>) defines the set of all <x>s satisfying a type. An object class, in the ODP meaning, represents the collection of objects that satisfy a given type. ODP makes the distinction template and class explicit. The class concept corresponds to the OMG extension concept; the extension of a type is the set of values that satisfy the type at a particular time. A subclass is a subset of a class. A subtype is therefore a predicate that defines a subclass. ODP subtype and subclass hierarchies are thus completely isomorphic.

A <x> template is the specification of the common features of a collection x in a sufficient detail that an x can be instantiated using it.

Types, classes, templates are needed for object, interface, and action.

3.2 The RM-ODP Enterprise Language

The definition of a language for each viewpoint describes the concepts and rules for specifying ODP systems from the corresponding viewpoint. The object concepts defined in each viewpoint language are specializations of those defined in the foundations part of RM-ODP

An enterprise specification is concerned with the purpose, scope and policies for the ODP system. Bellow, we summarize the basic enterprise concepts.

Community is the key enterprise concept. It is defined as a configuration of objects formed to meet an objective. The objective is expressed as a contract that specifies how the objective can be meet. A contract specifies an agreement governing part of the collective behavior of a set of objects. A contract specifies obligations, permissions and

prohibitions for objects involved. A contract specification may also include the specification of different roles engaged in the contract, the interfaces associated with the roles, quality of service attributes, indications of period of validity, behavior that invalidate the contract. The community specification also includes the environment contracts that state policies governing interactions of this community with its environment.

A role is a specification concept describing behavior. A role may be composed of several roles. A configuration of objects established for achieving some objective is referred to as a community. A role thus identifies behaviors to be fulfilled by the objects comprising the community.

A enterprise object is an object that fills one or more roles in a community.

A policy statement provides a additional behavioral specification. A community contract uses policy statements to separate behavioral specifications about roles.

A policy is a set of rules related to a particular purpose. A rule can be expressed as an obligation, a permission, or a prohibition. An ODP system consists of a set of enterprise objects. An enterprise object may be a role, an activity or a policy of the system.

4 Syntax Domain

We define a model to be the ODP enterprise viewpoint specification. That is, a set of enterprise objects formed to meet an objective. The objective is expressed as a

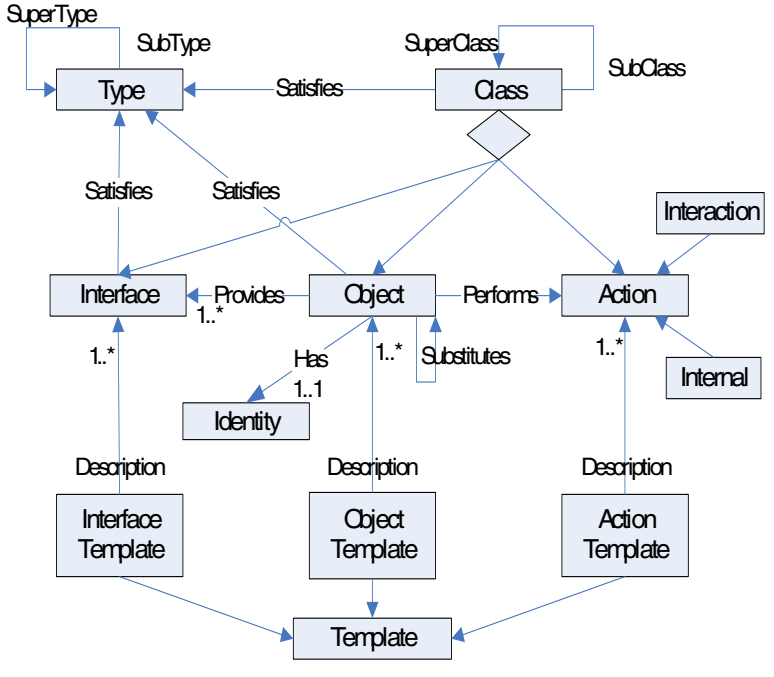


Fig. 1. RM-ODP Foundation Object Model

contract which specifies how the objective can be met A system can only be an instance of a single system model. Objects are instances of one ore more object templates; they may be of one or more types. The meta-model supports dynamic types.

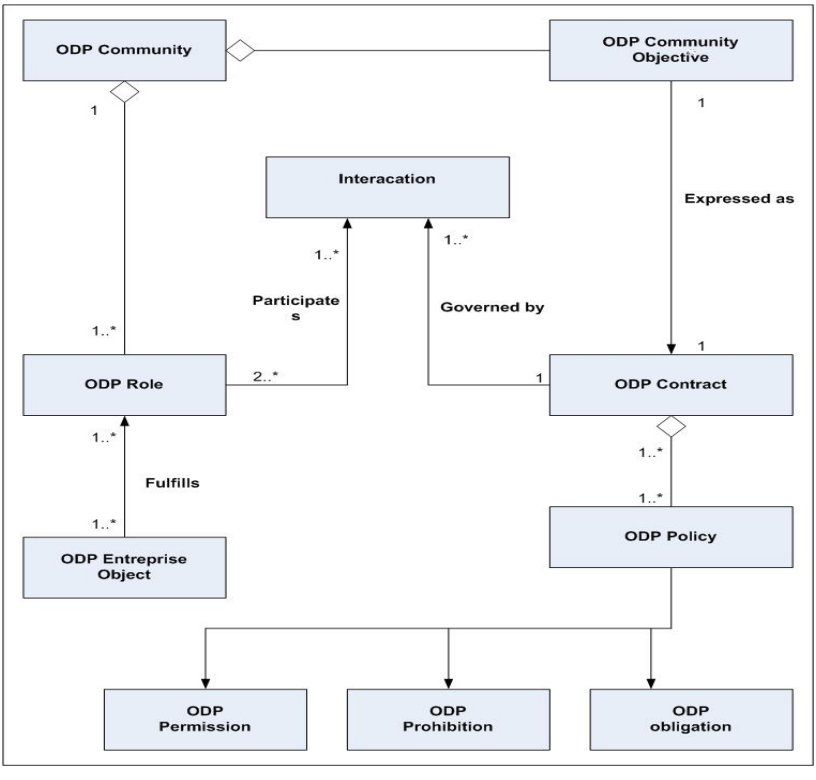


Fig. 2. Enterprise concepts

We define in this section the meta-models for the concepts presented in the previous section. Figure 1 defines the context free syntax for the core object concepts and figure 2, and figure 3 define the context free syntax for concepts in the enterprise language.

There are several context constraints between the syntax constructs. In the following we define some of these context constraints using OCL for the defined syntax.

Context m: Model inv:

m.Roles->includesAll(m.Roles.Source ->union(m.Roles.Target)
m.Roles->includesAll(m.ObjectTemplates.Roles)
m.Roles->includesAll(m.Interactiontemplate.roles)
m.Roles->includesAll(m.InterfaceTemplate.roles)

Context R: role inv
Community.enterepriseobject.role -> includes(self)
R.enterepriseobject -> includes(R.fulfils)

m.Types->includesAll(m.InteractionTemplates.Types-
>union(m.InterfaceTemplates.Types)- >union(m.InteractionTemplate.Target)

Context i : Interaction template inv :
r.role.inverse = r.Interactions.Roles.Source .inverse
and r.role.source = r.Interactions.Roles.Source .source
and r.role.source.inverse = r.Interactions.Roles.Source .inverse

Context o: Object Template inv:
eot (enterprise object template) is not parent of or child of itself
not (eot.parents ->includes(cot) or eot.children->includes(eot))

Context C: communityObject inv :
C.Enterepriseobject -> forall (E: Entereprise object |
E.instance -> inclaudesall(c.instance))
C.fulfilled_by -> size() >1

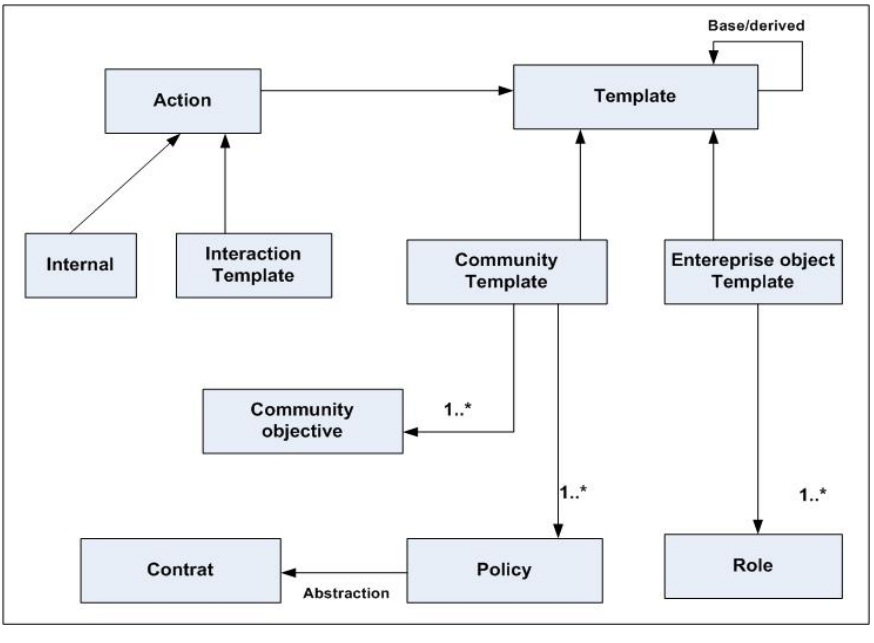


Fig. 3. Enterprise Concepts

5 Semantics Domain

The semantics of a UML model is given by constraining the relationship between a model and possible instances of that model. That is, constraining the relationship between expressions of the UML abstract syntax for models and expressions of the

UML abstract syntax for instances. The latter constitute the semantics domain and is integrated with the syntax domain within the same meta-model (fig1, fig.2 and fig3). This defines the UML context free abstract syntax. We give in the following a context constraint between the instances of semantics domain. These constrains are relatively simple but demonstrate the general principle.

We consider the concepts of subtype/supertype (RM-ODP 2-9.9) and subclass/ superclass (RM-ODP 2-9.10) as relations between types and classes correspondingly.

Context m : model inv

m.types-> forall(t1: Type, t2: Type | t2.subtype -> includes(t1) implies t1.valid_for.satisfies_type=t2)

m.types-> forall(t1: Type, t2: Type | t1.supertype ->includes(t2) implies t1.valid_for.satisfies_type=t2)

m.class-> forall(c1: Class, c2: Class | c2.subclass ->includes(c1) implies c2.associated_type.subtype) ->includes(c1.associated_type)

m.class-> forall(c1: Class, c2: Class | c1.superclass ->includes(c2) implies c2.associated_type.subtype) ->includes(c1.associated_type)

Context community inv :
Self.objective->size() = 1
Self.member_of ->includes (self.EnterepriseObject) -> size() > 1
Self.member_of ->includes (fulfils_role)
Self.role ->includes (sel. EnterepriseObject.Role) -> size() > 1
Self.member_of ->Forall(o:EO|self.o.role->notempty)

6 Meaning Function

The semantics for the UML-based language defined focuses on the relationship between a system model and its possible instances (systems). Both of the domains are defined using UML/OCL. The association between the instances of the two domains (syntax and semantics) is defined using the same UML meta-models. The OCL constraints complete the meaning function the context of UML [20, 31]. We give in the following some constraints which are relatively simple, but they demonstrate the general principle. We use the semantics of invariants as defined in [19].

Firstly there is a constraint relating to objects. It shows how inheritance relationships can force an object to be of many classes of its parents.

Context o: object inv:
The templates of o must be a sign template and all the parents of that template
o.of->exists(t | o.of=t->union(t.parents))

Secondly, there are four constraints which ensure that a model instance is a valid instance of the model it is claimed to be an instance of.

The first and second ensure that objects and interfaces are associated with templates known in the model.

Context s: system inv:

The model, that s is an instance of, includes all object templates that s.objects are instances of

s.of.ObjectTemplates->includesAll(s.Objects.of)

The model, that s is an instance of, includes all community templates that s.communitys are instances of

s.of.CommunityTemplates->includesAll(s.Communitys.of)

The third ensure that communities are associated with roles known in the model.

Context s: system inv:

The model, that s is an instance of, includes all the roles that s.community are instances of

s.of.roles ->includesAll(s.community.of)

The fourth constraint ensures that within the system cardinality constraints on roles are observed.

Context s: system inv:

s.community.of -> forAll(r | let community_in_s be
r.instances ->intersect (s.community) in
(r.upperBound -> notEmpty implies community_in_s ->size <= r.upperBound) and
coomunity_in_s->size >= r.upperbound)

7 Conclusion

We address in this paper the need of formal ODP viewpoint languages. Using the denotational meta-modeling semantics, we define in this paper the UML/OCL based syntax and semantics of a language for a fragment of ODP object concepts defined in the foundations part and in the enterprise viewpoint language. These concepts are suitable for describing and constraining the structure of ODP enterprise viewpoint specifications. We are applying the same denotataional semantics to define semantics for concepts characterizing dynamic behavior in other viewpoint languages.

References

1. ISO/IEC.: Basic Reference Model of Open Distributed Processing-Part1: Overview and Guide to Use, ISO/IEC CD 10746-1 (1994)
2. ISO/IEC.: RM-ODP-Part2: Descriptive Model, ISO/IEC DIS 10746-2 (1994)
3. ISO/IEC.: RM-ODP-Part3: Prescriptive Model, ISO/IEC DIS 10746-3 (1994)
4. ISO/IEC.: RM-ODP-Part4: Architectural Semantics, ISO/IEC DIS 10746-4 (July 1994)
5. OMG.: he Object Management Architecture, OMG (1991)
6. ISO/IEC.: ODP Type Repository Function, ISO/IEC JTC1/SC7 N2057 (1999)
7. ISO/IEC.: The ODP Trading Function, ISO/IEC JTC1/SC21 (1995)
8. Bouhdadi, M., et al.: An Informational Object Model for ODP Applications. Malaysian Journal of Computer Science 13(2), 21–32 (2000)
9. Spivey, J.M.: The Z Reference manual. Prentice Hall, Englewood Cliffs (1992)
10. IUT, SDL: Specification and Description Language, IUT-T-Rec. Z.100 (1992)

11. ISO/IUT.: LOTOS: A Formal Description Technique Based on the Temporal Ordering of Observational Behavior, ISO/IEC 8807 (1998)
12. Bowman, H., et al.: FDTs for ODP. Computer Standards & Interfaces Journal 17(5-6), 457–479 (1995)
13. Rumbaugh, J., et al.: The Unified Modeling Language. Addison Wesley, Reading (1999)
14. Rumpe, B.: A Note on Semantics with an Emphasis on UML. In: Demeyer, S., Bosch, J. (eds.) Object-Oriented Technology. ECOOP '98 Workshop Reader. LNCS, vol. 1543, pp. 167–188. Springer, Heidelberg (1998)
15. Evans, A., et al.: Making UML precise. In: Object Oriented Programming, Systems languages and Applications (OOPSLA'98), Vancouver, Canada, ACM Press, New York (1998)
16. Evans, A., et al.: The UML as a Formal Modeling Notation. In: Bézivin, J., Muller, P.-A. (eds.) UML 1998. LNCS, vol. 1618, pp. 349–364. Springer, Heidelberg (1999)
17. Warmer, J., Kleppe, A.: The Object Constraint Language: Precise Modeling with UML. Addison Wesley, Reading (1998)
18. Kent, S., et al.: A meta-model semantics for structural constraints in UML. In: Kilov, H., Rumpe, B., Simmonds, I. (eds.) Behavioral specifications for businesses and systems, ch. 9, Kluwer, Dordrecht (1999)
19. Evans, E., et al.: Meta-Modeling Semantics of UML. In: Kilov, H., Rumpe, B., Simmonds, I. (eds.) Behavioral specifications for businesses and systems, ch. 4. Kluwer, Dordrecht (1999)
20. Schmidt, D.A.: Denotational semantics: A Methodology for Language Development. Allyn and Bacon, Massachusetts (1986)
21. Myers, G.: The art of Software Testing. John Wiley &Sons, Chichester (1979)
22. Binder, R.: Testing Object Oriented Systems. Models. Patterns, and Tools. Addison-Wesley, Reading (1999)
23. Cockburn, A.: Agile Software Development. Addison-Wesley, Reading (2002)
24. Rumpe, B.: Agile Modeling with UML. In: Wirsing, M., Knapp, A., Balsamo, S. (eds.) RISSEF 2002. LNCS, vol. 2941, pp. 297–309. Springer, Heidelberg (2004)
25. Beck, K.: Column on Test-First Approach. IEEE Software 18(5), 87–89 (2001)
26. Briand, L.: A UML-based Approach to System testing. In: Gogolla, M., Kobryn, C. (eds.) UML 2001. LNCS, vol. 2185, pp. 194–208. Springer, Heidelberg (2001)
27. Rumpe, B.: Model-Based Testing of Object-Oriented Systems. In: de Boer, F.S., Bonsangue, M.M., Graf, S., de Roever, W.-P. (eds.) FMCO 2002. LNCS, vol. 2852, pp. 380–402. Springer, Heidelberg (2003)
28. Rumpe, B.: Executable Modeling UML. A Vision or a Nightmare?, In: Issues and Trends of Information technology management in Contemporary Associations, Seattle, Idea Group, London, pp. 697-701 (2002)
29. Bouhdadi, M., et al.: An UML-based Meta-language for the QoS-aware Enterprise Specification of Open Distributed Systems. In: IFIP Series, vol. 85, pp. 255–264. Springer, Heidelberg (2002)
30. Bouhdadi, M., et al.: M. A UML/OCL Denotational Semantics for ODP Structural Computational Concepts. In: First IEEE International Conference on Research Challenges in Information Science (RCIS'07), April 23-26, Ouarzazate, Morocco pp.259-264 (2007)
31. France, R., Kent, S., Evans, A., France, R.: What Does the Term Semantics Mean in the Context of UML. In: Moreira, A.M.D., Demeyer, S. (eds.) ECOOP 1999. LNCS, vol. 1743, pp. 34–36. Springer, Heidelberg (1999)

Component Specification for Parallel Coupling Infrastructure

J. Walter Larson[1,2] and Boyana Norris[1]

[1] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA
{larson,norris}@mcs.anl.gov
[2] ANU Supercomputer Facility, The Australian National University, Canberra ACT 0200 Australia

Abstract. *Coupled systems* comprise multiple mutually interacting subsystems, and are an increasingly common computational science application, most notably as multiscale and multiphysics models. Parallel computing, and in particular message-passing programming have spurred the development of these models, but also present a *parallel coupling problem* (PCP) in the form of intermodel data dependencies. The PCP complicates model coupling through requirements for the description, transfer, and transformation of the distributed data that models in a parallel coupled system exchange. Component-based software engineering has been proposed as one means of conquering software complexity in scientific applications, and given the compound nature of coupled models, it is a natural approach to addressing the parallel coupling problem. We define a software component specification for solving the parallel coupling problem. This design draws from the already successful Common Component Architecture (CCA). We abstract the parallel coupling problem's elements and map them onto a set of CCA components, defining a parallel coupling infrastructure toolkit. We discuss a reference implementation based on the Model Coupling Toolkit. We demonstrate how these components might be deployed to solve a relevant coupling problems in climate modeling.

1 Introduction

Multiphysics and multiscale models share one salient algorithmic feature: They involve interactions between distinct models for different physical phenomena. Multiphysics systems entail coupling of distinct interacting natural phenomena; a classic example is a coupled climate model, involving interactions between the Earth's atmosphere, ocean, cryosphere, and biosphere. Multiscale systems bridge distinct and interacting spatiotemporal scales; a good example can be found in numerical weather prediction, where models typically solve the atmosphere's primitive equations on multiple nested and interacting spatial domains. These systems are more generally labeled as *coupled systems*, and the set of interactions between their constituent parts are called *couplings*.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 55–68, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Though the first coupled climate model was created over 30 years ago [1], their proliferation has been dramatic in the past decade as a result of increased computing power.

On a computer platform with a single address space, coupling introduces algorithmic complexity by requiring data transformation such as integrid interpolation or time accumulation through averaging of state variables or integration of interfacial fluxes.

On a distributed-memory platform, however, the lack of a global address space adds further algorithmic complexity. Since *distributed data* are exchanged between the coupled model's constituent subsystems, their description must include a domain decomposition. If domain decompositions differ for data on source and target subsystems, the data traffic between them involves a communication schedule to route the data from source to destination. Furthermore, all data processing associated with data transformation will in principle involve explicit parallelism. The resultant situation is called the *parallel coupling problem* (PCP) [2, 3].

Myriad application-specific solutions to the PCP have been developed [4, 5, 6, 7, 8, 9, 10]. Some packages address portions of the PCP (e.g., the M×N problem—see [11]). Fewer attempts have been made to devise a flexible and more comprehensive solution [2, 12]. We propose here the development of a *parallel coupling infrastructure* toolkit, PCI-Tk, based on the *component-based software engineering* strategy defined by the Common Component Architecture Forum. We present a component specification for coupling, an implementation based on the Model Coupling Toolkit [2, 13, 14], and climate modeling as an example application.

2 The Parallel Coupling Problem

We begin with an overview of the PCP. For a full discussion readers should consult Refs. [2, 3].

2.1 Coupled Systems

A coupled system $\mathcal{M}$ comprises a set of N subsystem models called *constituents* $\{\mathcal{C}_1, \ldots, \mathcal{C}_N\}$. Each constituent $\mathcal{C}_i$ solves its equations of evolution for a set of *state variables* $\boldsymbol{\phi}_i$, using a set of *input variables* $\boldsymbol{\alpha}_i$, and producing a set of *output variables* $\boldsymbol{\beta}_i$. Each constituent $\mathcal{C}_i$ has a spatial domain Γ_i; its boundary $\partial\Gamma_i$ is the portion Γ_i exposed to other models for coupling.

The *state* $\mathbf{U}_i$ of $\mathcal{C}_i$ is the Cartesian product of the set of state variables $\boldsymbol{\phi}_i$ and the domain Γ_i, that is, $\mathbf{U}_i \equiv \boldsymbol{\phi}_i \times \Gamma_i$. The state $\mathbf{U}_i$ of $\mathcal{C}_i$ is computed from its current value and a set of *coupling inputs* $\mathbf{V}_i \equiv \boldsymbol{\alpha}_i \times \partial\Gamma_i$ from one or more other constituents in $\{\mathcal{C}_1, \ldots, \mathcal{C}_N\}$. *Coupling outputs* $\mathbf{W}_i \equiv \boldsymbol{\beta}_i \times \partial\Gamma_i$ are computed by $\mathcal{C}_i$ for use by one or more other constituents in $\{\mathcal{C}_1, \ldots, \mathcal{C}_N\}$. *Coupling* between between constituents $\mathcal{C}_i$ and $\mathcal{C}_j$ occurs if the following conditions hold:

1. Their computational domains overlap, that is, the *coupling overlap domain* $\Omega_{ij} \equiv \Gamma_i \cap \Gamma_j \neq \emptyset$.
2. They coincide in time.

3. Outputs from one constituent serve as inputs to the other, specifically (a) $\mathbf{W}_j \cap \mathbf{V}_i \neq \emptyset$ and/or $\mathbf{V}_j \cap \mathbf{W}_i \neq \emptyset$ or (b) the inputs $\mathbf{V}_i$ ($\mathbf{V}_j$) can be computed from the outputs $\mathbf{W}_j$ ($\mathbf{W}_i$).

In practice, the constituents are numerical models, and both Γ_i and $\partial\Gamma_i$ are discretized[1] by the discretization operator $\hat{\mathbf{D}}_i(\cdot)$, resulting in meshes $\hat{\mathbf{D}}_i(\Gamma_i)$ and $\hat{\mathbf{D}}_i(\partial\Gamma_i)$, respectively. Discretization of the domains Γ_i leads to definitions of state, input, and output vectors for each constituent $\mathcal{C}_i$: The *state vector* $\hat{\mathbf{U}}_i$ of $\mathcal{C}_i$ is the Cartesian product of the state variables $\boldsymbol{\phi}_i$ and the discretization of Γ_i; that is, $\hat{\mathbf{U}}_i \equiv \boldsymbol{\phi}_i \times \hat{\mathbf{D}}_i(\Gamma_i)$. The *input* and *output vectors* of $\mathcal{C}_i$ are defined as $\hat{\mathbf{V}}_i \equiv \boldsymbol{\alpha}_i \times \hat{\mathbf{D}}_i(\partial\Gamma_i)$ and $\hat{\mathbf{W}}_i \equiv \boldsymbol{\beta}_i \times \hat{\mathbf{D}}_i(\partial\Gamma_i)$, respectively.

The types of couplings in $\mathcal{M}$ can be classified as diagnostic or prognostic, explicit or implict. Consider coupling between $\mathcal{C}_i$ and $\mathcal{C}_j$ in which $\mathcal{C}_i$ receives among its inputs data from outputs of $\mathcal{C}_j$. *Diagnostic coupling* occurs if the outputs $\mathbf{W}_j$ used as input to $\mathcal{C}_i$ are computed *a posteriori* from the state $\mathbf{U}_j$. *Prognostic coupling* occurs if the outputs $\mathbf{W}_j$ used as input to $\mathcal{C}_i$ are computed as a forecast based on the state $\mathbf{U}_j$. *Explicit coupling* occurs if there is no overlap in space and time between the states $\mathbf{U}_i$ and $\mathbf{U}_j$. *Implicit coupling* occurs if there is overlap in space and time between $\mathbf{U}_i$ and $\mathbf{U}_j$, requiring a simultaneous, self-consistent solution for $\mathbf{U}_i$ and $\mathbf{U}_j$.

Consider explicit coupling in which $\mathcal{C}_i$ receives input from $\mathcal{C}_j$. The input state vector $\hat{\mathbf{V}}_i$ is computed from a *coupling transformation* $T_{ji} : \hat{\mathbf{W}}_j \rightarrow \hat{\mathbf{V}}_i$. The coupling transformation T_{ji} is a composition of two transformations: a *mesh transformation* $G_{ji} : \hat{\mathbf{D}}_j(\Omega_{ij}) \rightarrow \hat{\mathbf{D}}_i(\Omega_{ij})$ and a *field variable transformation* $F_{ji} : \boldsymbol{\beta}_j \rightarrow \boldsymbol{\alpha}_i$. Intergrid interpolation, often cast as a linear transformation, is a simple example of G_{ji}, but G_{ji} can be more general, such as a spectral transformation or a transformation betweeen Eulerian and Lagrangian representations. The variable transformation F_{ji} is application-specific, defined by the natural law relationships between $\boldsymbol{\beta}_j$ and $\boldsymbol{\alpha}_i$. In general, $G_{ji} \circ F_{ji} \neq F_{ji} \circ G_{ji}$; that is, the choice of operation order $G_{ji} \circ F_{ji}$ versus $F_{ji} \circ G_{ji}$ is up to the coupled model developer and is a source of coupled model uncertainty.

A coupled model $\mathcal{M}$ can be represented as a directed graph $\mathcal{G}$ in which the constituents are nodes and their data dependencies are directed edges. The connectivity of $\mathcal{M}$ is expressible in terms of the nonzero elements of the adjacency matrix $\mathbf{A}$ of $\mathcal{G}$. For a constituent's associated node, the number of incoming and outgoing edges corresponds to the number of *couplings*. If a node has only incoming (outgoing) edges, it is a sink (source) on $\mathcal{G}$, and this model may in principle be *run off-line*, using (providing) time history output (input) from (to) the rest of the coupled system. In some cases, a node may have two or more incoming edges, which may require *merging* of multiple outputs for use as input data. For a constituent $\mathcal{C}_i$ with incoming edges directed from $\mathcal{C}_j$ and $\mathcal{C}_k$, merging of data will be required to create $\hat{\mathbf{V}}_i$ if the following conditions hold:

[1] We will use the terms "mesh," "grid," "mesh points," and "grid points" interchangeably with the term "spatial discretization."

1. The constituents $\mathcal{C}_i$, $\mathcal{C}_j$, and $\mathcal{C}_k$ coincide in time.
2. The coupling domains Ω_{ij} and Ω_{ik} overlap, resulting in a *merge domain* $\Omega_{ijk} \equiv \Omega_{ij} \cap \Omega_{ik} \neq \emptyset$.
3. Shared variables exist among the fields delivered from $\mathcal{C}_j$ and $\mathcal{C}_k$ to $\mathcal{C}_i$, namely, $(\boldsymbol{\beta}_j \cap \boldsymbol{\alpha}_i) \cap (\boldsymbol{\beta}_k \cap \boldsymbol{\alpha}_i) \neq \emptyset$.

The time evolution of the coupled system is marked by *coupling events*, which either can occur predictibly following a *schedule* or can be *threshhold-triggered* based on some condition satisfied by the constituents' states. In some cases, the set of coupling events fall into a repeatable periodic schedule called a *coupling cycle*. For explicit coupling in which $\mathcal{C}_i$ depends on $\mathcal{C}_j$ for input, the time sampling of the output from $\mathcal{C}_j$ can come in the form of instantaneous values of $\hat{\mathbf{W}}_j$ or as a time integral of $\hat{\mathbf{W}}_j$, the latter used in some coupled climate models [15, 6].

2.2 Consequences of Distributed-Memory Parallelism

The discussion of coupling thus far is equally applicable to a single global address space or a distributed-memory parallel system. Developers of parallel coupled models confront numerous parallel programming challenges within the PCP. These challenges fall into two categories. *Coupled model architecture* encompasses the layout and allocation of the coupled model's resources and execution scheduling of its constituents. *Parallel data processing* is the set of operations necessary to accomplish data interplay between the constituents.

On distributed-memory platforms, the coupled-model developer faces a strategic decision regarding the mapping of the constituents to processors and the scheduling of their execution. Two main strategies exist—serial and parallel composition [16]. In a *serial composition*, all of the processors available are kept in a single pool, and the system's constituents each execute in turn using all of the processors available. In a *parallel composition* the set of available processors is divided into N disjoint groups called *cohorts*, and the constituents execute simultaneously, each on its own cohort. Serial composition has a simple conceptual design but can be a poor choice if the constituents do not have roughly the same parallel scalability; moreover, it restricts the model implementation to a single executable. Parallel composition offers the developer the option of sizing the cohorts based on their respective constituents' scalability; moreover, it enables the coupled model to be implemented as multiple executables. The chief disadvantage is that the concurrently executing constituents may be forced to wait for input data, causing cascading, hard-to-predict, and hard-to-control execution delays, which can complicate the coupled model's load balance. A third strategy, called *hybrid composition,* involves nesting one within the other to one or more levels (e.g., serial within parallel or vice versa). A fourth strategy, called *overlapping composition,* involves dividing the processor pool such that the constituents share some of the processors in their respective cohorts; this approach may be useful in implementing implicit coupling.

In a single global address space, description and transfer of coupling data are straightforward, and exchanges can be as simple as passing arguments through function interfaces. Standards for field data description (i.e., the $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$) and mesh description for the coupling overlap domains Ω_{ij} (i.e., their discretizations $\hat{\mathbf{D}}_i(\Omega_{ij})$) are sufficient. Distributed memory requires additional data description in the form of a *domain decomposition* $\mathbf{P}_i(\cdot)$, which splits the coupling overlap domain mesh $\hat{\mathbf{D}}_i(\Omega_{ij})$ and associated input and output vectors $\hat{\mathbf{V}}_i$ and $\hat{\mathbf{W}}_i$ into their respective local components across the processes $\{p_1, p_2, \ldots, p_{K_i}\}$, where K_i is the number of processors in the cohort associated with $\mathcal{C}_i$. That is, $\mathbf{P}_i(\hat{\mathbf{V}}_i) = \{\hat{\mathbf{V}}_i^1, \ldots, \hat{\mathbf{V}}_i^{K_i}\}$, $\mathbf{P}_i(\hat{\mathbf{W}}_i) = \{\hat{\mathbf{W}}_i^1, \ldots, \hat{\mathbf{W}}_i^{K_i}\}$, and $\mathbf{P}_i(\hat{\mathbf{D}}_i(\Omega_{ij})) = \{\hat{\mathbf{D}}_i^1(\Omega_{ij}), \ldots, \hat{\mathbf{D}}_i^{K_i}(\Omega_{ij})\}$.

Consider a coupling in which $\mathcal{C}_i$ receives input from $\mathcal{C}_j$. The transformation T_{ji} becomes a distributed-memory parallel operation, which in addition to its grid transformation G_{ji} and field transformation F_{ji} includes a third operation—*data transfer* H_{ji}. The order of composition of F_{ji}, G_{ji}, and H_{ji} is up to the model developer, and again the order of operations will affect the result. The data transfer H_{ji} will have less of an impact on uncertainties in the ordering of F_{ji} and G_{ji}, its main effect appearing in roundoff-level differences caused by reordering of arithmetic operations if computation is interleaved with the execution of H_{ji}. In addition, the model developer has a choice in the *placement* of operations, that is, on which constituent's cohort the variable and mesh transformations should be performed—the source, $\mathcal{C}_j$, the destination, $\mathcal{C}_i$, on a subset of the union of their cohorts, or someplace else (i.e., delegated to another constituent—called a *coupler* [4]—with a separate set of processes).

2.3 PCI Requirements

The abstraction of the PCP described above yields two observations: (1) the architectural aspects of the problem form a large decision space; and (2) the parallel data processing aspects of the problem are highly amenable to a generic software solution. Based on these observations, a parallel component infrastructure (PCI) must be modular, enabling coupled model developers to choose appropriate components for their particular PCP's. The PCI must provide decomposition descriptors capable of encapsulating each constituent's discretized domain boundary $\mathbf{P}_i(\hat{\mathbf{D}}_i(\partial\Gamma_i))$, its inputs $\mathbf{P}_i(\hat{\mathbf{V}}_i)$, and its outputs $\mathbf{P}_i(\hat{\mathbf{W}}_i)$. The PCI must provide communications scheduling for parallel data transfers and transposes needed to implement the H_{ij} operations for each model coupling interaction. Data transformation for coupling is an open-ended problem: Support for variable transformations F_{ij} will remain application-specific. The PCI should provide generic infrastructure for spatial mesh transformations G_{ij}, perhaps cast as a parallel linear transformation. Other desirable features of a PCI include spatial integrals for diagnosis of flux conservation under mesh transformations, time integration registers for time averaging of state data and time accumulation of flux data for implementing loose coupling, and a facility to merge output from multiple sources for input to a target constituent.

3 Software Components and the Common Component Architecture

Component-based software engineering (CBSE) [17, 18] is widespread in enterprise computing. A *component* is an atomic unit of software encapsulating some useful functionality, interacting with the outside world through well-defined interfaces often specified in an *interface definition language*. Components are composed into applications, which are executed in a runtime *framework*. CBSE enables software reuse, empowers application developers to switch between subsystem implementations for which multiple components exist, and can dramatically reduce application development time. Examples of commercial CBSE approaches include COM, DCOM, JavaBeans, Rails, and CORBA. Alas, they are not suitable for scientific applications for two reasons: unreasonably high performance cost, especially in massively parallel applications, and inability to describe scientific data adequately (e.g., they do not support complex numbers).

The Common Component Architecture (CCA) [19, 20] is a CBSE approach targeting high-performance scientific computing. CCA's approach is based on explicit descriptions of *software dependencies* (i.e., caller/callee relationships). CCA component interfaces are known as *ports*: *provides* ports are the interfaces implemented, or provided, by a component, while *uses* ports are external interfaces whose methods are called, or used, by a component. Component interactions follow a *peer component model* through a connection between a uses ports and a provides port of the same type to establish a caller/callee relationship (Figure 1 (a)). The CCA specification also defines some special ports (e.g., GoPort, essentially a "start button" for a CCA application). CCA interfaces as well as port and component definitions are described in a SIDL (*scientific interface definition language*) file, which is subsequently processed by a language interoperability tool such as Babel [21] to create the necessary interlanguage glue code. CCA meets performance criteria associated with high-performance computing, and typical latency times for intercomponent calls between components executing on the same parallel machine are on the order of a virtual function call between same-language components; times for interlanguage component interactions are slightly more, but within 1–2 orders of magnitude of typical MPI latency times.

The port connection and mediation of calls is handled by a CCA-compliant framework. Each component implements a `SetServices()` method where the component's uses and provides ports are registered with the framework. At runtime, uses ports are connected to provides ports, and a component can access the methods of a port through a `getPort()` method. A typical port connection diagram for a simple application (which will be discussed in greater detail in Section 6) is shown in Figure 1 (b).

CCA technology has been applied successfully in many application areas including combustion, computational chemistry, and Earth sciences. CCA's language interoperability approach has been leveraged to create multilingual bindings for the Model Coupling Toolkit (MCT) [22], the coupling middleware used by

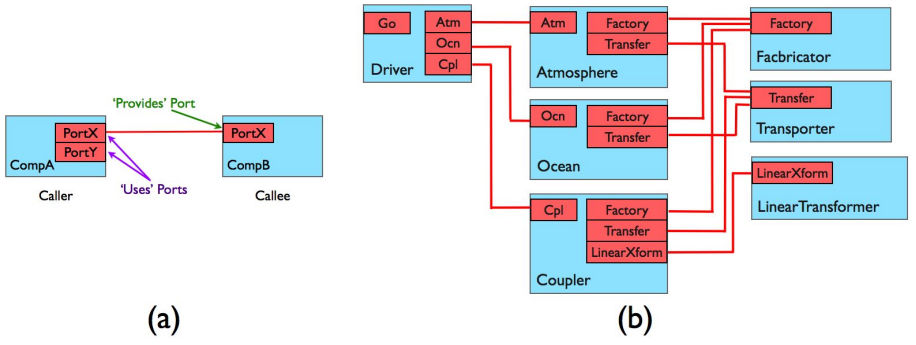


Fig. 1. Sample CCA component wiring diagrams: (a) generic port connection for two components; (b) a simple application composed from multiple components.

the Community Climate System Model (CCSM), and a Python implementation of the CCSM coupler. The work reported here will eventually be part of the CCA Toolkit.

4 PCI Component Toolkit

Our PCI specification is designed to address the majority of the requirements stated in Section 2.3. Emphasis is on the parallel data processing part of the PCP, with a middleware layer immediately above MPI that models can invoke to perform parallel coupling operations. A highly modular approach that separates concerns at this low level maximizes flexibility, and this bottom-up design allows support for serial and parallel compositions and multiple executables. We have defined a standard API for distributed data description and constituent processor layout. These standards form a foundation for an API for parallel data transfer and transformation. Below we outline the PCI API and the component and port definitions for PCI-Tk.

4.1 Data Model for Coupling

In our specification, the objects for data description are the SpatialGrid ($\hat{\mathbf{D}}_i(\Gamma_i)$ and $\hat{\mathbf{D}}_i(\partial\Gamma_i)$), the FieldData (the input and output vectors $\hat{\mathbf{V}}_i$ and $\hat{\mathbf{W}}_i$) and the GlobalIndMap (the domain decomposition $\mathbf{P}_i$). This approach assumes a 1-1 mapping between the elements of the spatial discretization and the physical locations in the field data definition. The domain decomposition applies equally to both. For example, a constituent $\mathcal{C}_i$ spread across a cohort of K_i processors, each processor (say, the kth) will have its own SpatialGrid to describe $\hat{\mathbf{D}}_i^k(\partial\Gamma_i)$, and FieldData instantiations to describe its local inputs and outputs $\hat{\mathbf{V}}_i^k$ and $\hat{\mathbf{W}}_i^k$, respectively. In the interest of generality and minimal burden to PCI implementers, we have adopted explicit *virtual linearization* [11,2,23,24,25,26] as our

index and mesh description standard. Virtual linearization supports decomposition description of multidimensional index spaces and meshes, both structured and unstructured. We have adopted an explicit, segmented domain decomposition [2, 11] of the linearized index space.

Data transfer within PCI requires a description of the constituents' cohorts and communications schedules for interconstituent parallel data transfers and intracohort parallel data redistributions. Mapping of constituent processor pools is described by the CohortRegistry interface, which provides MPI processor ID ranks for a constituent's processors within its own MPI communicator and a union communicator of all model cohorts. The CohortRegistry provides lookup services necessary for interconstituent data transfers. Our PCI data model provides two descriptors for the transfer operation H_{ji}: The TransferSched API is an interface that encapsulates interconstituent parallel data transfer scheduling; that is, it contains all of the information necessary to execute all of the MPI point-to-point communication calls needed to implement the transfer.

The data transformation part of the PCI requires data models for linear transformations and for time integration. The LinearTransform encapsulates the whole transformation from storage of transformation coefficients to communications scheduling required to execute the parallel transformation. The TimeIntRegister describes time integration and averaging registers required for loose coupling in which state averages and flux integrals are exchanged periodically for incremental application.

In the SIDL PCI API, all of the elements of the data model are defined as *interfaces*; their implementation as classes or otherwise is at the discretion of the PCI developer.

4.2 PCI-Tk Components

Data Description

The Fabricator Component. The Fabricator creates objects used in the interfaces for all the coupling components, along with their associated service methods. It also handles overall MPI communicator management This component has a single provides port, Factory, on which all of the create/destroy, query, and manipulation methods for the coupling data objects reside.

Data Transfer. Data under transfer by our PCI interfaces is described by our FieldData specification.

The Transporter Component. The Transporter performs one-way parallel data transfers such as the data routing between source and destination constituents, with communications scheduling described by our TransferSched interface. It has one provides port, Transfer, on which methods for both blocking (`PCI_Send()`, `PCI_Recv()`) and nonblocking (`PCI_ISend()`, `PCI_IRecv()`) parallel data transfers are implemented, making it capable of supporting both serial and parallel compositions.

The Transposer Component. The Transposer performs two-way parallel data transfers such as data redistribution within a cohort, or two-way data traffic between constituents, with communications scheduling defined by the TransposeSched interface. It has one provides port, Transpose, that implements a data transpose function `PCI_Transpose()`.

Data Transformation. The data transformation components in PCI-Tk act on FieldData inputs and, unless otherwise noted, produce outputs described by the FieldData specification.

The LinearTransformer Component. The LinearTransformer performs parallel linear transformations using user-defined, precomputed transform coefficients. It has a single provides port, LinearXForm, that implements the transformation method `PCI_ApplyLinearTransform()`.

The TimeIntegrator Component. The TimeIntegrator performs temporal integration and averaging of FieldData for a given constituent, storing the ongoing result in a form described by the TimeIntRegister specification. It has a single provides port, TimeInt, that implements methods for time averaging and integration, named `PCI_TimeIntegral()` and `PCI_TimeAverage()`, respectively. Users can retrieve time integrals in FieldData form from a query method associated with the TimeIntRegister interface.

The SpatialIntegrator Component. The SpatialIntegrator performs spatial integrals of FieldData on its resident SpatialGrid. It has a single provides port, SpatialInt, that offers methods `PCI_SpatialIntegral()` and `PCI_SpatialAverage()` that perform multifield spatial integrals and averages, respectively. This port also has methods for performing simultaneously paired multifield spatial integrals and averages; here *pairing* means that calculations for two different sets of FieldData on their respective resident SpatialGrid objects are computed. This functionality enables efficient, scalable diagnosis of conservation of fluxes under transformation from source to target grids.

The Merger Component. The Merger merges data from multiple sources that have been transformed onto a common, shared SpatialGrid. It has a single provides port, Merge, on which merging methods reside, including `PCI_Merge2()`, `PCI_Merge3()`, and `PCI_Merge4()` for merging of data from two, three, and four sources, respectively. An additional method `PCI_MergeIn()` supports higher-order and other user-defined merging operations.

5 Reference Implementation

We are using MCT to build a reference implementation of our PCI specification. MCT provides a data model and library support for parallel coupling. Like the specification, MCT uses virtual linearization to describe multidimensional index

Table 1. Correspondence between PCI Data Interfaces and MCT Classes

Functionality	PCI Interface	MCT Class
Mesh Description $\hat{\mathbf{D}}_i\Gamma_i$, $\hat{\mathbf{D}}_i(\partial\Gamma_i)$	SpatialGrid	GeneralGrid
Field Data $\hat{\mathbf{U}}_i$, $\hat{\mathbf{V}}_i$, $\hat{\mathbf{W}}_i$	FieldData	AttrVect
Domain Decomposition $\mathbf{P}_i$	GlobalIndMap	GlobalSegMap
Constituent PE Layouts	CohortRegistry	MCTWorld
One-Way Parallel Data Transfer Scheduling H_{ij}	TransferSched	Router
Two-Way Parallel Data Transpose Scheduling H_{ij}	TransposeSched	Rearranger
Linear Transformation G_{ij}	LinearTransform	SparseMatrix SparseMatrixPlus
Time Integration Registers	TimeIntRegister	Accumulator

Table 2. Correspondence between PCI Ports and MCT Methods

Component / Port	MCT Method
Fabricator / Factory	Create, destroy, query, and manipulation methods for GeneralGrid, AttrVect, GlobalSegMap, MCTWorld, Router, Rearranger, SparseMatrix, SparseMatrixPlus, and Accumulator
Transporter / Transfer	Transfer Routines `MCT_Send()`, `MCT_Recv()`, `MCT_ISend()`, `MCT_IRecv()`,
Transposer / Transpose	`Rearrange()`
LinearTransform / LinearXForm	SparseMatrix-AttrVect Multiply `sMatAvMult()`
SpatialIntegrator / Spatial Integral	`SpatialIntegral()` and `SpatialAverage()`
TimeIntegrator / TimeIntegral	`accumulate()`
Merger / Merge	`Merge()`

spaces and grids. MCT's Fortran API is described in SIDL, and Babel has been used to generate multilingual bindings [22], with Python and C++ bindings and example codes available from the MCT Web site. The data model from our PCI specification maps readily onto MCT's classes (see Table 1).

The port methods are implemented in some cases through direct use (via glue code) of MCT library routines, and at worst via lightweight wrappers that perform minimal work to convert port method arguments into a form usable by MCT. Table 2 shows in broad terms how the port methods are implemented.

6 Deployment Examples

We present three examples from climate modeling in which PCI-Tk components could be used to implement parallel couplings. the field of coupled climate modeling. In each example, the system contains components for physical subsystems and a *coupler*. The coupler handles the data transformation, and the models interact via the coupler purely through data transfers—a *hub-and-spokes* architecture [15].

The MCT toy climate coupling example comprises atmosphere and ocean components that interact via a coupler that performs intergrid interpolation, and computes application-specific variable transformations such as computation of interfacial radiative fluxes. It is a single executable application; the atmosphere, ocean, and coupler are procedures invoked by the MAIN driver application. It is a parallel composition; parallel data transfers between the cohorts are required. A CCA wiring diagram of this application using PCI-Tk components is shown in Figure 1 (b). The driver component with the Go port signifies the single executable, and this component has uses ports labeled Atm, Ocn, and Cpl implemented as provides ports on the atmosphere, ocean, and coupler components, respectively. The PCI-Tk data model elements used in the coupling are created and managed by the Fabricator, via method calls on its Factory port. The parallel data transfer traffic between the physical components and the coupler are implemented by the Transporter component via method calls on its Transfer port. A LinearTransform component is present to implement interpolation between the atmosphere and ocean grids; the coupler performs this task via method calls on its LinearXform port.

The *Parallel Climate Model (PCM) example* [6] shown in Figure 2 (a) is a single executable and a serial composition. A driver coordinates execution of the individual model components. Since the models run as a serial composition, coupling data can be passed across interfaces, and transposes performed as needed; thus there is a Transpose component rather than a Transfer component. The coupler in this example performs the full set of transformations found in PCM: intergrid interpolation with the LinearTransform; diagnosis of flux conservation under interpolation with the SpatialIntegrator; time integration of flux and

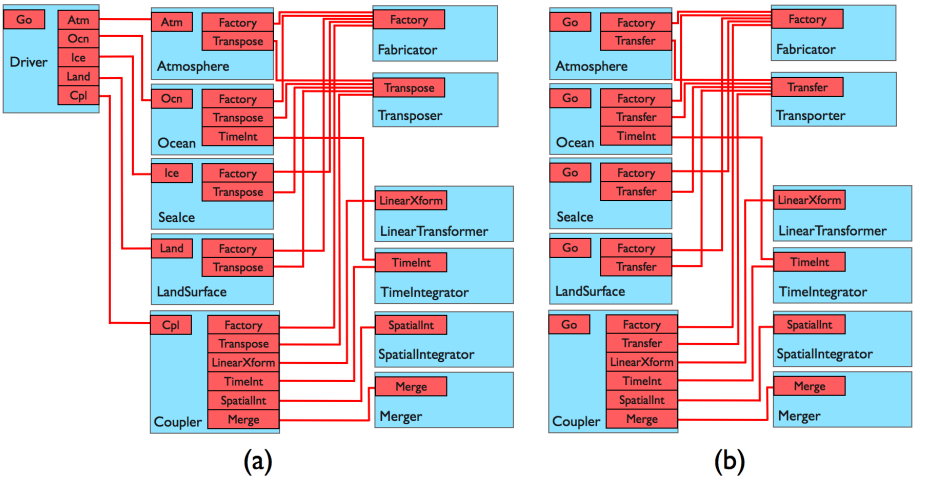


Fig. 2. CCA wiring diagrams for a two coupled climate model architectures: (a) PCM, with serial composition and single executable; (b) CCSM, with parallel composition and multiple executables

averaging of state data using the TimeIntegrator; and merging of data from multiple sources with the Merger. The TimeIntegrator is invoked by both the ocean and coupler components because of the loose coupling between the ocean and the rest of PCM; the atmosphere, sea-ice, and land-surface models interact with the couple hourly, but the ocean interacts with the coupler once per model day. The coupler integrates the hourly data from the atmosphere, land, and sea-ice that will be passed to the ocean. The ocean integrates its data from each timestep over the course of the model day for delivery to the coupler.

The *CCSM example* is a parallel composition. Its coupling strategy is similar to that in PCM in terms of the parallel data transformations and implementation of loose coupling to the ocean. CCSM uses a *peer communciation* model, however, with each of the physical components communicating in parallel with the coupler. These differences are shown in Figure 2 (b). The atmosphere, ocean, sea-ice, land-surface, and coupler are separate executables and have Go ports on them; and the parallel data transfers are implemented by the Transfer component rather than the Transpose component.

7 Conclusions

Coupling and the PCP are problems of central importance as computational science enters the age of multiphysics and multiscale models. We have described the theoretical underpinnings of the PCP and derived a core set of PCI requirements. From these requirements, we have formulated a PCI component interface specification that is compliant with the CCA, a component approach suitable for high-performance scientific computing—a *parallel coupling infrastructure toolkit* (PCI-Tk). We have begun a reference implementation based on the Model Coupling Toolkit MCT. Use-case scenarios indicate that this approach is highly promising for climate modeling applications, and we believe the reference implmentation will perform approximately as well as MCT does: the component overhead introduced by CCA has been found to be acceptably low in other application studies [19]; and our own performance studies on our Babel-generated C++ and Python bindings for MCT show minimial performance impact (at most a fraction of a percent versus the native Fortran implementation [22]).

Future work includes completing the reference implementation and a thorough performance study; prototyping of applications using the MCT-based PCI-Tk; modifying the specification if necessary; and exploring alternative PCI component implementations (e.g., using mesh and field data management tools from the DOE-supported Interoperable Technologies for Advanced Petascale Simulations (ITAPS) center [27].

Acknowledgements. This work is primarily a product of the Center for Technology for Advanced Scientific Component Software (TASCS), which is supported by the US Department of Energy (DOE) Office of Advanced Scientific Computing Research through the Scientific Discovery through Advanced Computing Program, . Argonne National Laboratory is managed for the DOE by UChicago Argonne LLC under supported by the DOE under contract DE-AC02-06CH11357.

The ANU Supercomputer Facility is funded in part by the Australian Department of Education, Science, and Training through the Australian Partnership for Advanced Computing (APAC).

References

1. Manabe, S., Bryan, K.: Climate calculations with a combined ocean-atmosphere model. Journal of the Atmospheric Sciences 26(4), 786–789 (1969)
2. Larson, J., Jacob, R., Ong, E.: The Model Coupling Toolkit: A new Fortran90 toolkit for building multi-physics parallel coupled models. Int. J. High Perf. Comp. App. 19(3), 277–292 (2005)
3. Larson, J.W.: Some organising principles for coupling in multiphysics and multiscale models. Preprint ANL/MCS-P1414-0207, Mathematics and Computer Science Division, Argonne National Laboratory (2006)
4. Bryan, F.O., Kauffman, B.G., Large, W.G., Gent, P.R.: The NCAR CSM flux coupler. NCAR Tech. Note 424, NCAR, Boulder, CO (1996)
5. Jacob, R., Schafer, C., Foster, I., Tobis, M., Anderson, J.: Computational design and performance of the Fast Ocean Atmosphere Model. In: Alexandrov, V.N., Dongarra, J.J., Juliano, B.A., Renner, R.S., Tan, C.J.K. (eds.) ICCS 2001. LNCS, vol. 2073, pp. 175–184. Springer, Heidelberg (2001)
6. Bettge, T., Craig, A., James, R., Wayland, V., Strand, G.: The DOE Parallel Climate Model (PCM): The Computational Highway and Backroads. In: Alexandrov, V.N., Dongarra, J.J., Juliano, B.A., Renner, R.S., Tan, C.J.K. (eds.) ICCS 2001. LNCS, vol. 2073, pp. 148–156. Springer, Heidelberg (2001)
7. Drummond, L.A., Demmel, J., Mechose, C.R., Robinson, H., Sklower, K., Spahr, J.A.: A data broker for distirbuted computing environments. In: Alexandrov, V.N., Dongarra, J.J., Juliano, B.A., Renner, R.S., Tan, C.J.K. (eds.) ICCS 2001. LNCS, vol. 2073, pp. 31–40. Springer, Heidelberg (2001)
8. Valcke, S., Redler, R., Vogelsang, R., Declat, D., Ritzdorf, H., Schoenemeyer, T.: OASIS4 user's guide. PRISM Report Series 3, CERFACS, Toulouse, France (2004)
9. Hill, C., DeLuca, C., Balaji, V., Suarez, M., da Silva, A.: The ESMF Joint Specification Team: The architecture of the earth system modeling framework. Computing in Science and Engineering 6, 18–28 (2004)
10. Toth, G., Sokolov, I.V., Gombosi, T.I., Chesney, D.R., Clauer, C.R., Zeeuw, D.D., Hansen, K.C., Kane, K.J., Manchester, W.B., Oehmke, R.C., Powell, K.G., Ridley, A.J., Roussev, I.I., Stout, Q.F., Volberg, O., Wolf, R.A., Sazykin, S., Chan, A., Yu, B., Kota, J.: Space weather modeling framework: A new tool for the space science community. Journal of Geophysical Research 110, A12226 (2005)
11. Bertrand, F., Bramley, R., Bernholdt, D.E., Kohl, J.A., Sussman, A., Larson, J.W., Damevski, K.B.: Data redistribution and remote method invocation for coupled components. Journal of Parallel and Distributed Computing 66(7), 931–946 (2006)
12. Joppich, W., Kurschner, M.: MpCCI - a tool for the simulation of coupled applications. Concurrency and Computation: Practice and Experience 18(2), 183–192 (2006)
13. Jacob, R., Larson, J., Ong, E.: M x N communication and parallel interpolation in ccsm3 using the Model Coupling Tookit. Int. J. High Perf. Comp. App. 19(3), 293–308 (2005)
14. The MCT Development Team: Model Coupling Toolkit (MCT) web site (2007), `http://www.mcs.anl.gov/mct/`

15. Craig, A.P., Kaufmann, B., Jacob, R., Bettge, T., Larson, J., Ong, E., Ding, C., He, H.: cpl6: The new extensible high-performance parallel coupler for the community climate system model. Int. J. High Perf. Comp. App. 19(3), 309–327 (2005)
16. Foster, I.: Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering. Addison Wesley, Reading, Massachusetts (1995)
17. Szyperski, C.: Component Software: Beyond Object-Oriented Programming. ACM Press, New York (1999)
18. Heineman, G.T., Council, W.T.: Component Based Software Engineering: Putting the Pieces Together. Addison-Wesley, New York (1999)
19. Bernholdt, D.E., Allan, B.A., Armstrong, R., Bertrand, F., Chiu, K., Dahlgren, T.L., Damevski, K., Elwasif, W.R., Epperly, T.G.W., Govindaraju, M., Katz, D.S., Kohl, J.A., Krishnan, M., Kumfert, G., Larson, J.W., Lefantzi, S., Lewis, M.J., Malony, A.D., Mclnnes, L.C., Nieplocha, J., Norris, B., Parker, S.G., Ray, J., Shende, S., Windus, T.L., Zhou, S.: A component architecture for high-performance scientific computing. Int. J. High Perf. Comp. App. 20(2), 163–202 (2006)
20. CCA Forum: CCA Forum web site (2007), `http://cca-forum.org/`
21. Dahlgren, T., Epperly, T., Kumfert, G.: Babel User's Guide. CASC, Lawrence Livermore National Laboratory. version 0.9.0 edn. (January 2004)
22. Ong, E.T., Larson, J.W., Norris, B., Jacob, R.L., Tobis, M., Steder, M.: Multilingual interfaces for parallel coupling in multiphysics and multiscale systems. In: Shi, Y. (ed.) ICCS 2007. LNCS, vol. 4487, pp. 924–931. Springer, Heidelberg (2007)
23. Lee, J.-Y., Sussman, A.: High performance communication between parallel programs (Appears with the Proceedings of IPDPS 2005). In: Proceedings of 2005 Joint Workshop on High-Performance Grid Computing and High-Level Parallel Programming Models (HIPS-HPGC 2005), Apr 2005, IEEE Computer Society Press, Los Alamitos (2005)
24. Sussman, A.: Building complex coupled physical simulations on the grid with InterComm. Engineering with Computers 22(3–4), 311–323 (2006)
25. Jones, P.W.: A user's guide for SCRIP: A spherical coordinate remapping and interpolation package. Technical report, Los Alamos National Laboratory, Los Alamos, NM (1998)
26. Jones, P.W.: First and second-order conservative remapping schemes for grids in spherical coordinates. Mon. Wea. Rev. 127, 2204–2210 (1999)
27. Interoperable Technologies for Advanced Petascale Simulation Team: ITAPS web site (2007), `http://www.scidac.gov/math/ITAPS.html`

Real-Time Navigation for a Mobile Robot Based on the Autonomous Behavior Agent

Lu Xu, Liguo Zhang, and Yangzhou Chen

School of Electronic Information & Control Engineering,
Beijing University of Technology, Beijing, 100022, China
flysand4@gmail.com

Abstract. Autonomous behavior agent includes the robot most behaviors which are designed using hierarchical control method to guarantee their real time performance for real time navigation in response to different situation perceived. The process of robot time navigation based on the autonomous behavior agent inly includes three vio s. The sensing behavior translates the configuration space that the robot and obstacles exist in into 2D Cartesian Grid by Quadtree method. The path planning behavior designs the sub-goals given the global map, start and goal points by improved D* Lite Algorithm. And the obstacle avoidance behaivor replans the path between two adjacent sub-goals when the environment changes. It is able to replan faster than planning from scratch since it modifies its previous search results locally and enables robots adapt to the dynamic environment. The simulation results that are reported show that the mobile robot navigation method is efficient and feasible.

Keywords: obile robot; -time navigation; dynamic environment; path planning; autonomous agent.

1 Introduction

Real time navigation compasses the ability of the robot to act based on its knowledge and sensor values so as to reach its goal positions as efficiently and as reliably as possible, given partial knowledge about its dynamic environment and a goal position or series of positions. Behavior control [1-3] is an important and new intelligent navigation control method, which combines all information robot apperceiving and repository together to the global environment map. The basic behavior is the cell of feeling or execution. The robot behaviors are divided into two major types: Reflexive Behaviors and Conscious Behaviors. The reflexive behaviors are stimulus-response, and do not need computation process. The fundamental attribute of the reflexive paradigm is that all actions are accomplished through behaviors. Rodney Brooks' [1] subsumption architecture is the typical reflexive paradigm. The conscious behaviors are deliberative, which join the planning into the reflexive paradigm, making the robot has the memory and reasoning ability.

One kind of approaches in the global environment based on the known map is Road Map Path Planning. It is a decomposition of the robot's configuration space based specifically on obstacle geometry. The typical method is Voronoi Diagram [4].

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 69 – 80, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Given a particular planned path via Voronoi Diagram planning, the robot with range sensors, can follow a Voronoi edge in the physical world using simple control rules that match those used to create the Voronoi diagram. But it has an important weakness in the case of limited range localization. Since this algorithm maximizes the distance between the robot and objects in the environment, any short-range sensor on the robot will be in danger of failing to sensor its surroundings. Another kind of approaches is Cell Decomposition Path Planning. NF1 [5] is an efficient and simple-to-implement technique for finding routes in fixed-size cell arrays. The algorithm simply employs wavefront expansion from the goal position outward, marking for each cell its distance to the goal cell [6]. Its transform is breadth-first search implemented in the constrained space of an adjacency array [7]. The fundamental cost of the fixed decomposition approach is memory. For a large environment, even when sparse, this grid must be represented in its entirety. Practically, due to the falling cost of computer memory, this disadvantage has been mitigated in recent years.

One kind of approaches in the local environment based on the partial information is D* (Dynamic A*) [8-9], which developed by Stentz and demonstrated on an autonomous vehicle for the US DARPA Unmanned Ground Vehicle Demo II project. It achieves a large speedup over repeated A* [10] searches by modifying previous search results locally. D* has been extensively used on real robots. It is currently also being integrated into Mars Rover prototypes and tactical mobile robot prototypes for urban reconnaissance [11]. Incremental search technique is guaranteed to be as good as the quality of the solutions obtained by replanning from scratch. M. Likhachev developed an incremental version of A* [12-14], Lifelong Planning A* (LPA*), combining ideas from the algorithm literature and the artificial intelligence literature. LPA* uses heuristics to focus the search and always finds a shortest path for the current edge costs. However, it achieves a substantial speedup over A* because it reuses those parts of the previous search tree that are identical to the new search tree. D* Lite Algorithm [15] which is an incremental version of the heuristic search method A* and combines ideas from Lifelong Planning A* and Focussed D*. It implements the same navigation strategy as Focussed D* but is algorithmically simpler. It always moves the robot on a shortest path from its current grid to the goal grid, and replans the shortest path when the edge costs change.

In this paper, it describes a new robot behaivor agent, which on the one side gives programming users a very flexible system, which is modular, extensible, transparent and portable for different robot platforms, and on the other side allows a fast development of applications. It mainly includes sensing behavior, path planning behavior and obstacle avoidance behavior. The rest of the paper is organized as follows. In section 2, the autonomous behavior agent is presented, and sensing behavior, path planning behavior and obstacle avoidance behavior are introduced in section 3, 4, 5. In section 6, the operation of the method with examples and present simulation results are given. Section 7 draws conclusions and future extensions.

2 Autonomous Behavior Agent

Autonomous behavior agent fulfills the following common demands: modularity, namely different modules of the agent must be functional independent and exchangeable;

extensibility, namely the agent must be easily extensible with new modules; transparency, namely an exchange or a modification of a single module must be transparent to the other modules; efficiency, namely the agent must be able to run in real-time on the robot navigation.

Autonomous behavior agent includes the robot most behaviors which are designed using hierarchical control method to guarantee their real time performance in response to different context, situation and states perceived. Also, those behaviors satisfy the constraint conditions of driving system, and they include terrain features and prior knowledge of motion control to enhance their reactivity.

The agent interacts with the user and determines the goals and order of execution, as shown in Fig.1. The overall agent is comprised of a set of modules, each module connected to other modules via inputs and outputs. The system is closed, meaning that the input of every module is the output of one or more modules. The Path Planning behavior is engaged, planning the global sub-goals. It uses all available global information in non real time to identify the right sequence of local actions for the robot. The obstacle avoidance behavior takes the desired heading and adapts it to the obstacles. It is responsible for the activation of behaviors based on information it receives from the sensors that is different from its pre-information. It is particularly useful in the mobile robot with dynamic maps that become more accurate as the robot moves. It combines the versatility of responding to environmental changes and new goals with the fast response of a tactical executive tier and behaviors.

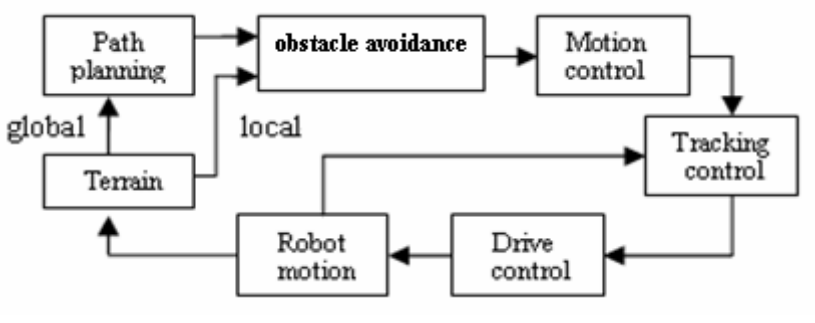


Fig. 1. The configuration of autonomous behavior agent

The process of robots real-time navigation mainly includes the following three steps: firstly, translate the configuration space that the robot and obstacles exist in into 2D Cartesian Grid by Quadtree method. Secondly, design the sub-goals given the global map, start and goal points. Finally, replan the path between two adjacent sub-goals when the environment changes.

3 Sensing Behavior

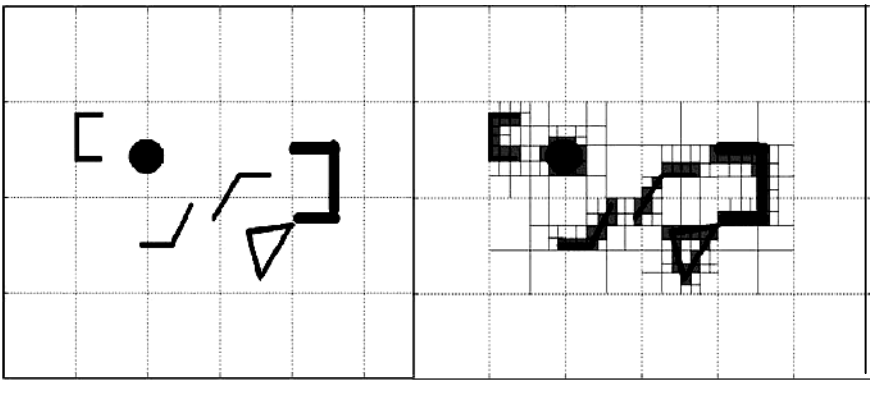
The robot is equipped with stereo cameras, mounted on a motion-averaging mast. A laser range-finder, mounted on the top of the robot, is a device which uses a laser beam in order to determine the distance to an opaque object with incredible accuracy.

It works by sending a laser pulse in a narrow beam towards the object and measuring how long it takes for the pulse to bounce off the target and return to the sender.

The physical space that a robot and obstacles exist in can be thought as the configuration space (Cspace). It is a data structure that allows the robot to specify the position. A good Cspace representation reduces the number of dimensions that a planner has to contend with. Six dimensions data can accurately express the robot position: Descartes coordinate (x, y, z) specify the location, Euller （Φ θ γ specify the position. Six degrees of freedom is less often used for a mobile ground robot in most cases for planning and only need (x, y) to express position.

Superimpose 2D Cartesian Grid on the configuration space. If there is any obstacle in the area contained by a grid element, that element is marked occupied. If sensors inspect the obstacle in front of it falls into part of the grid, but not all of it, the Quadtree method divides the element into four smaller grids. If the obstacle doesn't fill a particular sub-element, it does another recursive division of that element into four more sub-elements, until the grid's occupied proportion reaches a given coefficient, as shown in Fig.2.

In order to remove the digitization bias (an obstacle falls into even the smallest portion of a grid element, the whole element is marked occupied), the general grid space method divides the Cspace into baby-sized grid elements. This means a high storage cost, and a high number of nodes to consider. The Quadtree method avoids the waste of structure space as well as removes the digitization bias.



(a): General Cspace (b): Quadtree Cspace

Fig. 2. The configuration space

In this paper, it is to assume for path-planning purposes that the robot is in fact holonomic, simplifying the process tremendously. And further assume that the robot is a point. It can reduce the Cspace for mobile robot navigation to a 2D representation with just x- and y- axes. At the same time, it must inflate each obstacle by the size of the robot's radius to compensate.

4 Path Planning and Replanning Behavior

The robot path planning has a lot of optimum performances, such as time optimization, route optimization, energy consumption optimization etc. This paper chooses the route optimization as the algorithm performance. For the sake of finding out a path between two points in the map, if at least exist a path, and the search time is the least, incremental heuristics algorithm is the most suitable method. Most search methods replan from scratch, that is, solve the search problems independently. Incremental search technique share with case-based planning, they find solutions to series of similar search tasks much faster than is possible by solving each search task from scratch. Improved D* Lite algorithm is a kind of heuristic search method. It makes use of the heuristic function to search the direction along which it can easier find out the target and produce the optimum result, and takes into account that the heuristics change when the robot moves and the goal of the search thus changes..

The pseudo code as shown in Fig.3 uses the following notation. S denotes the finite set of grid of the graph. $s_{start} \in$ S denotes the current grid of the robot, initially the start grid, and $s_{goal} \in$ S denotes the goal grid. Succ(s) $\subseteq$S denotes the set of successors of s $\in$ S. Similarly, Pred(s) $\subseteq$S denotes the set of predecessors of s $\in$ S. 0 c(s, s')$\leq\infty$ denotes the cost of moving from s to s' $\in$ Succ(s). The heuristics h(s, s') estimate the distance between grid s and s'. D* Lite requires that the heuristics are nonnegative and satisfy h(s,s') $\leq$ c(s, s') for all grid s $\in$ S, s' Succ(s) and h(s, s'') $\leq$ h(s, s')+h(s', s'') for all grid s, s', s'' $\in$ S. And it maintains two kinds of estimates of the start distance g*(s) of each grid s: a g-value g(s) and an rhs-value rhs(s). The rhs-value of a grid is based on the g-values of its predecessors and thus potentially better informed than them. It always satisfies the following relationship [15]:

$$rhs(s) = \begin{cases} 0 & if, s = s_{start} \\ \min_{s' \in \Pr ed(s)} (g(s') + c(s', s)) & otherwise \end{cases} \quad (1)$$

A vertex is called locally consistent iff its g-value equals its rhs-value, otherwise it is either overconsistent iff g(s) > rhs(s) or underconsistent iff g(s) < rhs(s). The priority queue OPEN always holds exactly the inconsistent states, these are the states that need to be updated and made consistent, until there is no state in the queue with a key value less than that of the start state. The key value k(s) {3} is a vector with two components:

$$k(s) = [k_1(s) = \min(g(s), rhs(s)) + h(s), k_2(s) = \min(g(s), rhs(s))] \quad (2)$$

k(s) is smaller than or equal to a key k'(s), denoted by $k(s) \leq k^{'}(s)$, iff either $k_1(s) < k_1^{'}(s)$ or $k_1(s) = k_1^{'}(s)$ and $k_2(s) \leq k_2^{'}(s)$. $k_1(s)$ corresponds directly to the f-values f(s)=g(s)+h(s) used by A* because both the g-values and rhs-values of D* Lite correspond to the g-values of A* and the h-values of D* Lite correspond to the h-values of A*. $k_2(s)$ corresponds to the g-values of A*. D* Lite expands vertices in

```
procedure Initialize()
{1}open = close= [];
{2}rhs( s_goal )=0; g( s_goal )=∞;
{3} k(s) = [k1(s), k2(s)];
{4}Insertopen( s_goal , k(s));
procedure Updatevertex(s)
{5}if s ≠ s_goal
{6}   if s ∈ open
{7}      rhs(s) = min(rhs,rhs(s));
{8}   elseif s ∈ close & g(s) ≠ min(rhs,rhs(s))
{9}      Insertopen(s);
{10}     Removeclose(s);
{11}  elseif s ∉ open & s ∉ close
{12}     g(s)= ∞;
{13}     Insertopen(s);
procedure Reupdatevertex(s)
{14}if s ≠ s_goal
{15}  if s ∈ open
{16}     rhs(s) = rhs;
{17}  elseif s ∈ close
{18}     rhs(s) = rhs;
{19}     Insertopen(s);
{20}     Removeclose(s);
{21}  elseif s ∉ open & s ∉ close
{22}     g(s)= ∞;
{23}     Insertopen(s);
procedure Computepath()
{24}while( s_top ≠ s_start ||
           rhs( s_start )≠g( s_start ))
{25}if g( s_top ) > rhs( s_top )
{26}  g( s_top )=rhs( s_top );
{27}  for all s ∈ Pred( s_top )
{28}     rhs=rhs( s_top )+1;
{29}     Updatevertex(s);
{30}  end
{31}  Insertclose( s_top );
{32}  Removeopen( s_top );
{33}else
{34}  g( s_top )=∞;
{35}  for all s ∈ Pred( s_top ) ∪ { s_top }
{36}     if s ≠ s_goal
{37}      rhs(s)=min(g(s')+1); (s' ∈ Succ(s))
{38}     Reupdatevertex(s);
procedure Main()
{39}while ( s_start ≠ s_goal )
{40} s_start = arg min (g(s')); (s' ∈ Succ(s))
{41}Move to s_start
{42}scan graph
{43}if edge changed
{44}  close( s_start )=[];
{45}  for all s ∈ Pred( s_start )
{46}     rhs(s)=min(g(s'))+1; (s' ∈ Succ(s))
{47}     Reupdatevertex(s);
```

Fig. 3. The improved D* Lite Algorithm

the order of increasing k_1-values and vertices with equal k_1-values in order of increasing k_2-values. This is similar to A* that expands vertices in the order of increasing f-values and vertices with equal f-values that are on the same branch of the search tree in order of increasing g-values.

Fig.4 presents a path planning and replanning example on a simple grid world. In this example it has an eight-connected grid where black cells represent obstacles and white cells represent free space. The cell marked "start" denotes the position of a robot navigating this environment towards the cell marked "goal". The cost of moving from one cell to any non-obstacle neighboring cell is one. The heuristic function is the larger of the x axis and y axis distances from the current cell to the cell occupied by the robot. The cells expanded are shown in blue. The replanning paths are shown as dark lines.

In an actual implementation, Initialize() only needs to initialize a vertex when it encounters it during the search and thus does not need to initialize all vertices up

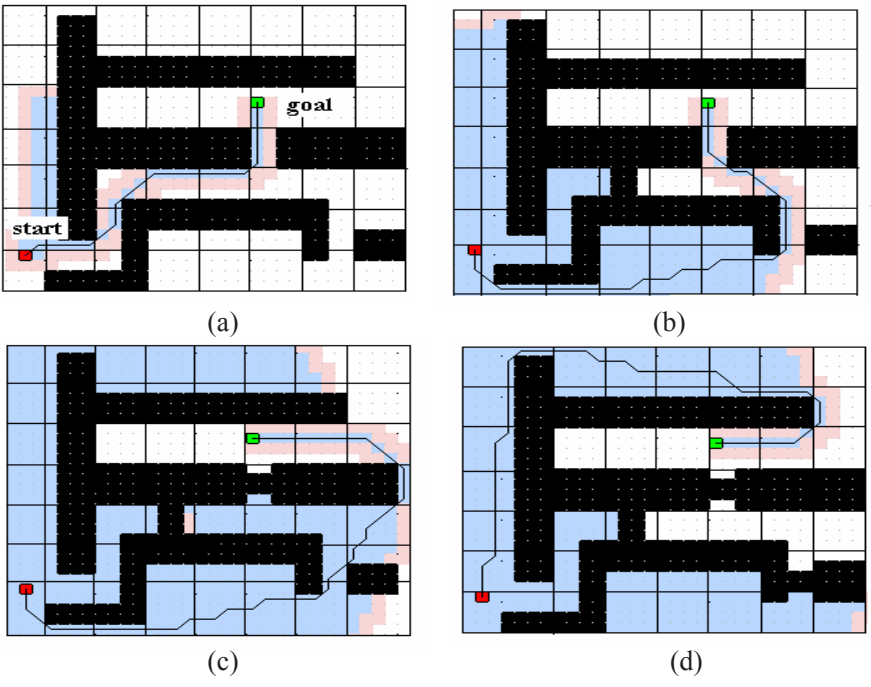


Fig. 4. Path planning & Replanning

front. D* Lite waits for changes in edge costs {42}. If some edge costs have changed, it calls Reupdatevertex() {45-46} to update the rhs-valus and keys of the vertices potentially affected by the changed edge costs as well as their membership in the priority queue if they become locally consistent or inconsisitent, and finally recalculates a shortest path. It turns out that D* Lite expands a vertex at most twice, namely at most once when it is underconsistent and at most once when it is overconsistent [14].

Theorem 1. Within any particular execution of Computepath function once a state is expanded as overconsistent or underconsistent it can never be expanded again.

Proof: Suppose a state s is selected for expansion as overconsistent or underconsistent for the first time during the execution of the Compputepath function. Then it is removed from OPEN and inserted into CLOSED. It can then never be inserted into OPEN again unless the Computepath function exits since any state that is about to be inserted into OPEN is checked against membership in CLOSED. Because only the states from OPEN are selected for expansion, s can therefore never be expanded second time.

5 Obstacle Avoidance Behavior

When avoiding obstacle on-line, the mobile robot only obtains the certain scope Cspace information by sensors. Making use of the information of this space is the prior condition of obstacle avoidance.

Suppose that the inspected district is a rectangle and divide this space according with the sensing behavior. Every grid is seen as a vector where the size equals $v_{i,j}$ and the direction equals $\beta_{i,j}$. $v_{i,j}$ represents the density of obstacle in that grid. The bigger the value is, the more the danger is.

$$v_{i,j} = c_{i,j} \cdot (a - b \cdot d_{i,j}) \tag{3}$$

Where,

$$c_{i,j} = \begin{cases} 1, obstalce \\ 0, \quad non-obstacle \end{cases};$$

$d_{i,j}$ is the distance between the grid center and the robot position;

a, b are the positive constants and $a - b \cdot d_{\max} = 0$.

$$\beta_{i,j} = tg^{-1} \frac{y_j - y_0}{x_i - x_0} \tag{4}$$

Where,

$(x_i \quad y_j)$ is the coordinate of the grid center;

$(x_0 \quad y_0)$ is the coordinate of the robot position.

Disperse the two dimension rectangle by angle, and translate it into a one- dimension polar coordinate histogram. The one-dimension polar coordinate histogram is composed of n fan-shaped sectors where each angle is α, n=180/α. Every fan-shaped sector k (k=1,2, … , n) has the value h_k which represents the cost of moving along this direction k·α.

$$h_k = \sum_{i,j} v_{i,j} \tag{5}$$

The relation that grid subjects to the fan-shaped sector k is:

$$k = \text{INT}(\beta_{i,j} / \alpha) \tag{6}$$

Through equation (3-6), the grids in the inspected district have been mapped into the relevant fan-shaped sectors.

The candidate direction k_w is the robot motion direction whose value is lower than the threshold in the polar coordinate histogram. Make use of evaluation function g(k_w) to compare each candidate direction, and the direction k_l whose g(k_l) is the least is the robot motion direction.

$$g(k_w) = \mu_1 \cdot \Delta(k_w, k_t) + \mu_2 \cdot \Delta(k_w, INT(\frac{\theta_n}{\alpha})) + \mu_3 \cdot \Delta(k_w, k_{n-1}) \tag{7}$$

Where,

k_t is the target direction;

INT $\left(\frac{\theta_n}{\alpha}\right)$ is the robot's current motion direction;

k_{n-1} is the previous velocity direction;

△ represents the difference between the two items in the bracket;

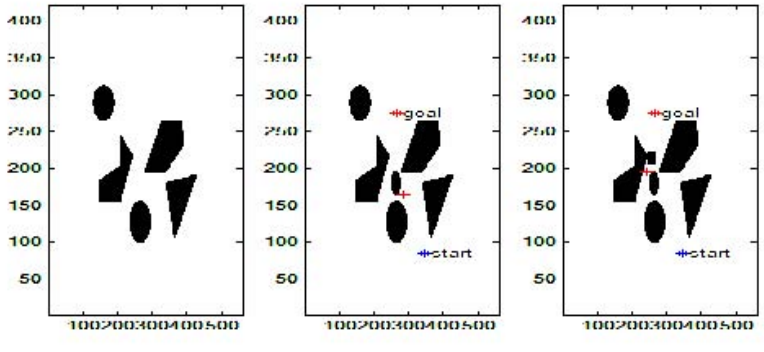
μ_1 μ_2 μ_3 are harmonious genes.

Each item in the evaluation function g(k_w) equals the sub- behavior of obstacle avoidance: No.1 item is the moving to goal sub- behavior, No.2 item is the smoothing track sub- behavior, No.3 item is the contenting control restriction sub- behavior. When we emphasize one behavior, its corresponding harmonious gene becomes big. For example, when we emphasize the moving to goal sub-behavior, we let $\mu_1 > \mu_2 + \mu_3$.

6 Simulation Results

Fig.5 shows an example of robot navigation in dynamic environment. The robot first got the 5(a) map. It planned the path given the start and goal point. Then it moved along the path and stopped at the "+" point of 5(b) map because it sensed the environment changed. So it needed to replan the path given the current point and the goal point. When it moved at the "+" point of 5(c) map, the environment changed again, so it stopped and replanned the path. The final moving track is shown in Fig.6. The cells with "+" are the path that robot moved and the cells with "×" are expanded cells.

In the process of mobile robot navigation, we must consider the restrictions of the dynamics and kinematics. Such as the robot can't be regarded as a point, the speed and acceleration have peak value limitations, and it exists the minimum turn



(a) (b) (c)

Fig. 5. The dynamic environment

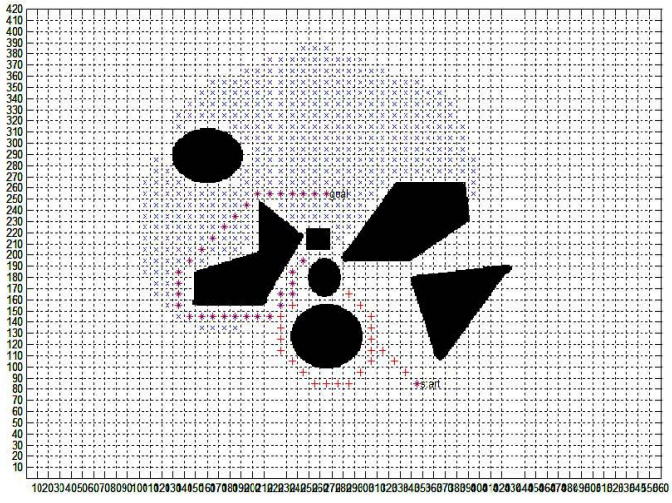


Fig. 6. The robot navigation in dynamic environment

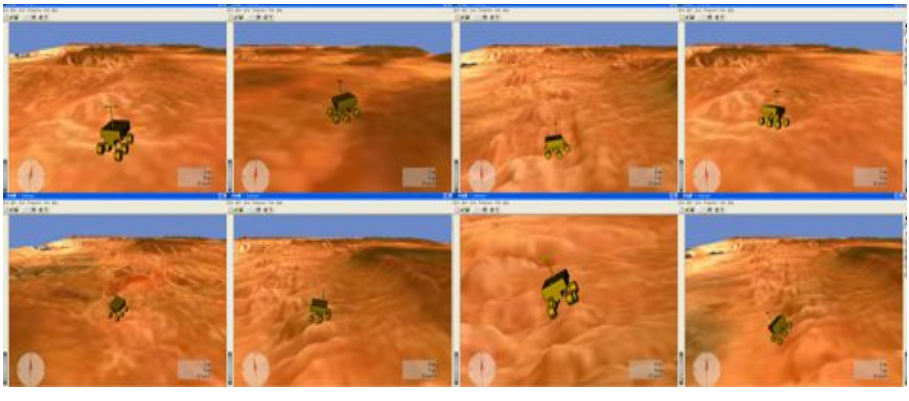


Fig. 7. The robot real time navigation based on the behavior agent

radius, etc. These restrictions are prior conditions of mobile robot safety motion. The simulation results of robot real-time navigation based on the behavior agent are shown in Fig.7.

7 Conclusions

The robot real-time navigation for goal-directed navigation in known environment has been studied extensively, but in unknown environment has been studied less frequently. It needs the methods that repeatedly determine a shortest path from the current robot coordinates to the goal coordinates while the robot moves along the path. Incremental search methods typically solve dynamic shortest path problems, that is,

path problems where shortest paths between a given start and goal point have to be determined repeatedly as the topology of a graph or its edge costs change.

This paper presents the robot navigation method based on the autonomous behavior agent. The sensing behavior translates the configuration space that the robot and obstacles exist in into 2D Cartesian Grid by Quadtree method. The path planning behavior designs the sub-goals given the global map, start and goal points by improved D* Lite Algorithm. And the obstacle avoidance behaivor replans the path between two adjacent sub-goals when the environment changes.

Acknowledgment

This work is supported partly by National Natural Science Foundation of China (No.60374067). The authors would like to acknowledge all the members from Research Center of Behavior-based Technology & Intelligent Control in Beijing University of Technology, for their help in the study.

References

1. Brooks, R.A.: A Robust Layered Control System for A Mobile Robot, IEEE Journal of Robotics And Automation, RA-2(1) (March 1986)
2. Samer, A., Masoud, Masoud, A.A.: Constrained Motion Control Using Vector Potential Fields. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans 30(3) (2000)
3. Khatib, 0.: A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation. IEEE Journal of Robotics and Automation RA. 3(1) (1987)
4. Choset, H., Walker, S.: Sensor-Based Exploration: Incremental Construction of the Hierarchical Generalized Voronoi Graph. The International Journal of Robotics Research 19, 126–148 (2000)
5. Latombe, J-C., Barraquand, J.: Robot Motion Planning: A Distributed Presentation Approach. International Journal of Robotics Research 10, 628–649 (1991)
6. Jacobs, R., Canny, J.: Planning Smooth Paths for Mobile Robots. In: Proc. IEEE Conference on Robotics and Automation, pp. 2–7. IEEE Press, Los Alamitos (1989)
7. Russell, S., Norvig, P.: Artificial Intelligence, a Modern Approach. Prentice-Hall, Englewood Cliffs (1995)
8. Stentz, A.: Optimal and efficient path planning for partially-known environments. In: Proc. Int. Conf. Robot. Autom., pp. 3310–3317 (1994)
9. Stentz, A.: The Focussed D* Algorithm for Real Time Replanning. In: Proc. Int. Joint Conf. Artificial Intelligence, pp. 1652–1659 (1995)
10. Pearl, J.: Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley, Reading, MA (1985)
11. Hebert, M., et al.: Experiments with Driving Modes for Urban Robots. In: Proc. SPIE Mobile Robots, pp. 140–149 (1999)
12. Likhachev, M.: Anytime Dynamic A*: An Anytime, Replanning Algorithm. In: Proc. Int. Conf. Automated Planning and Scheduling (2005)

13. Koening, S., Likhachev, M.: Incremental A*, Advances in Neural Information Processing Systems 14. MIT Press, Cambridge, MA (2002)
14. Likhachev, M.: Search-based Planning for Large Dynamic Environments. PhD thesis, CMU (2005)
15. Koenig, S.: Fast Replanning for Navigation in Unknown Terrain. IEEE Transactions on Robotics 21(3) (2005)

Concurrent Subsystem-Component Development Model (CSCDM) for Developing Adaptive E-Commerce Systems

Liangtie Dai[1] and Wanwu Guo[2]

[1] Institute of Human Resources Management, Jinan University
Guangzhou, Guangdong Province, China
ojwqg@jnu.edu.cn
[2] School of Computer and Information Science, Edith Cowan University
2 Bradford Street, Mount Lawley, Western Australia 6050, Australia
w.guo@ecu.edu.au

Abstract. The waterfall and incremental models are widely used for guiding E-commerce system development. In some cases where clients demand a quick solution to maximise their business benefit, these models are not fully fit in directing such projects. This is because the client's prioritisation of system requirements is determined by immediate business benefit whereas the developer's is based on the long-term system usability and reliability. The concurrent subsystem-component development model (CSCDM) proposed in this paper is an alternative approach for guiding system development, especially for client-driven business systems. Its waterfall-based framework is easy to follow and understand by the developers. Its component-based stream only iterates over the Implementation, Testing, and Deployment stages. Since the design for the overall system and prototypes is done at the same stage, all the prototypes can be fully or partly absorbed into the corresponding system groups, and/or modified to fit into the system design. The local iterations realise the business needs separately using prototyping without interference with the progression of the overall system development, which ensures the quick deployment of a prototype. The case of WCE Pizza online system proves that CSCDM is an effective approach to guide the development of client-driven E-commerce systems.

Keywords: Electronic commerce, Waterfall model, Incremental model, Prototyping, Concurrent subsystem-component development model (CSCDM).

1 Introduction

In supervising projects on electronic commerce system development carried out by the final-year university students, the waterfall model is commonly recommended to students for guiding their system development. This is because the duration of a student E-commerce project is usually one semester and thus the size of the project is relatively small. This model is especially useful for developing a small E-commerce system whose requirements have been well defined in the commencement and whose deployment date is explicitly set.

In a few cases where clients demand a quick solution to maximise their business benefit, the incremental model is used for guiding such E-commerce system development.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 81 – 91, 2007.
© Springer-Verlag Berlin Heidelberg 2007

However, experience shows that this model is still not fully capable of dealing with such projects. This is because the client's prioritisation of system requirements may be different from that of the developer's. The former is determined by immediate business benefit whereas the latter is based on the long-term system usability and reliability.

In this paper, the author proposes a new model to handle such client-driven system development for E-commerce applications. It is named as the concurrent subsystem-component development model (CSCDM), which combines the system-based waterfall model and component-based prototyping together with local increments to deliver the functional components of the system in succession based on the client's prioritisation of system requirements, unlike other approaches that realise the subsystems in sequence.

2 E-Commerce System Architectures and Development Models

E-commerce system is a kind of distributed systems and usually utilises the client-server architecture to facilitate business activities [1,2,3]. A client is software residing in a user's machine for requesting services through communication network (such as the Internet) from the service provider, or the server. A server is a collection of implementations handling some specific services on behalf of a collection of clients and usually residing in another powerful machine (two-tiered architecture) or other machines (multi-tiered architecture) [3,4].

No matter whether it is a two-tiered or multi-tiered system, in terms of functionality, any client-server system consists of three logical layers: presentation (user interface), processing (business logic), and data (database) (Fig. 1). The user-interface level contains all that is necessary to directly interface the user, and thus is implemented in clients. Processing level contains the core functionality for establishing client-server communication, executing business procedure, and interacting with database for data management. Data level contains the actual data and programs that maintain the data [1,3].

These three logical layers can be placed in different physical entities (machines). If only two machines are used, the presentation logic is normally placed in the user's machine whereas the processing and data logics are installed on the server side. In system categorisation, presentation is called as the front-end (sub)system and both processing and data together are named as the back-end (sub)system (Fig. 1). When three physical parties host the three logical layers, the data layer is usually placed into an independent machine unknown as database. Although we can still use front-end and back-end systems to describe this three-tiered organisation, it is often categorised as the user interface (sub)system, business processing (sub)system, and database (sub)system, respectively, in system categorisation (Fig. 1).

Development of such client-server E-commerce system follows the Systems Development Life Cycle (SDLC), which is presented as different system development models. Although variations exist in different models, the five constituents – Analysis, Design, Implementation, Testing, and Deployment – form the backbone of many models [5,6,7].

The simple but yet widely used model for developing small E-commence systems is known as the waterfall model illustrated in Fig. 2 [5,6,8]. The five constituents are

formed as a chain of sequential stages, with each stage being completed before the commencement of the next stage. This model is useful in developing a small E-commerce system that has well-defined system requirements as the explicit input, and expects a client-server application with predicated behaviours as its output. This system development model does not tolerate any major change in system requirements during development because such a change may lead to the restart of the whole procedure.

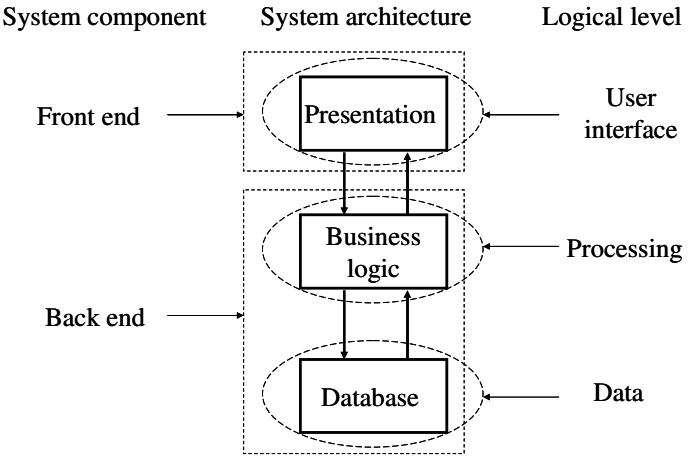


Fig. 1. Client-server system architecture and logical layering

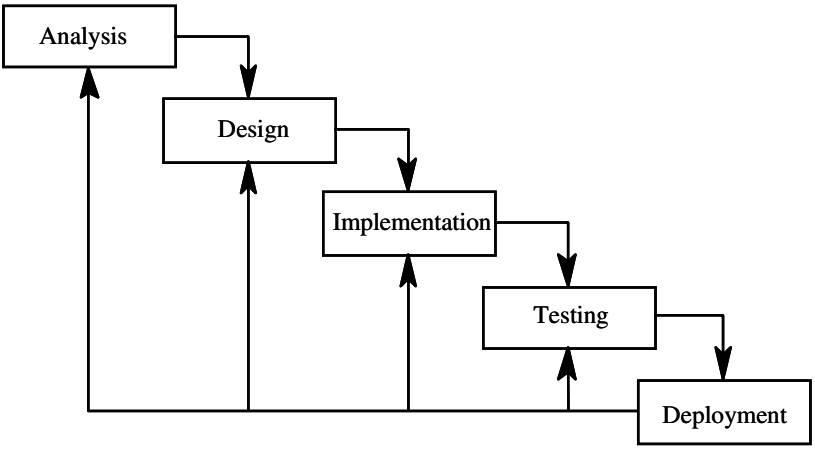


Fig. 2. The waterfall system development model

The goal of the incremental model is to design and deliver to the client a minimal subset of the whole system that is still a usable system [6,9]. The development will continue to iterate through the whole life cycle with additional increments (Fig. 3).

This model performs the waterfall in overlapping sections, producing usable functionality earlier. Each subsequent release of the system adds function to the previous release, until all designed functionality has been implemented. This model implies that a high level of inheritance and reusability must be maintained during the development.

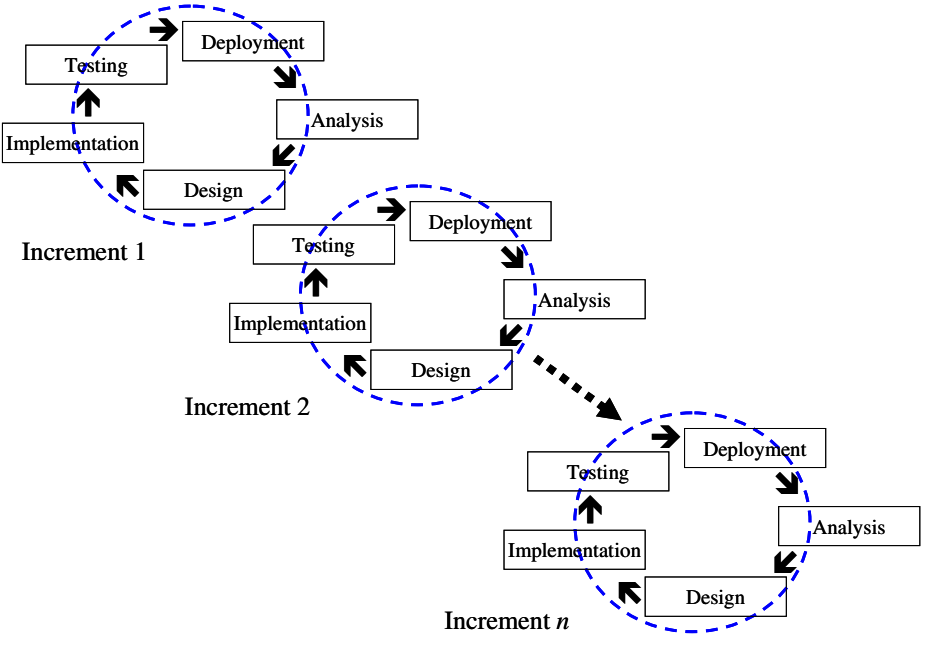
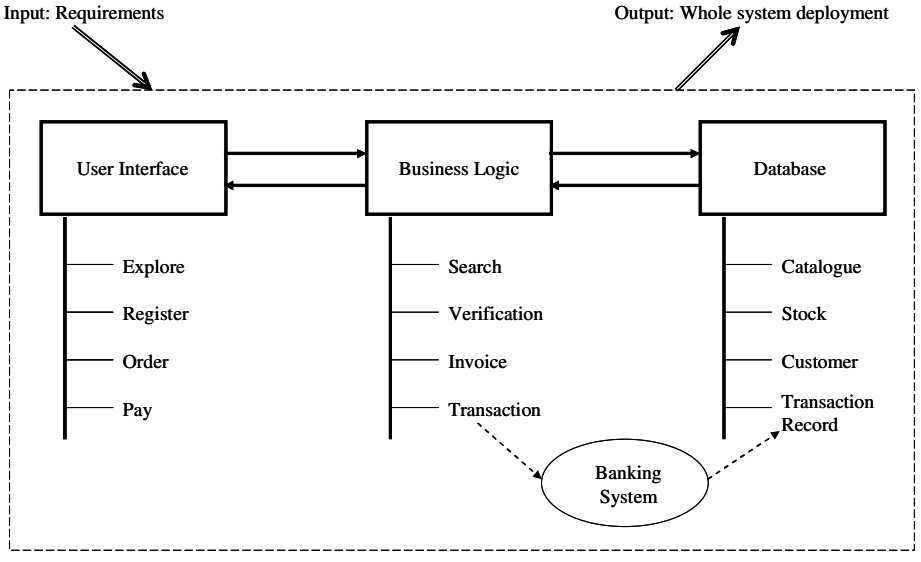


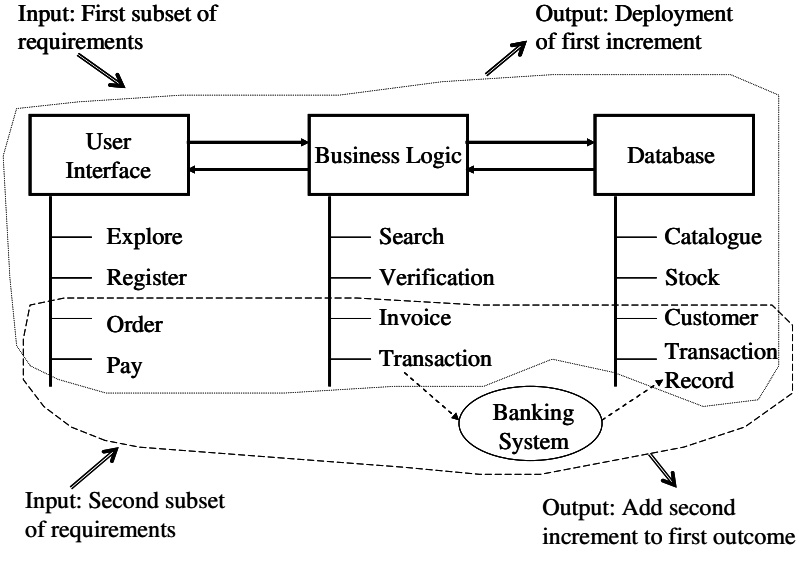
Fig. 3. The incremental system development model

Both waterfall and incremental models adopt system or subsystem oriented approaches for system development. The fundamental difference between these two models is that the waterfall delivers the whole system once as the final outcome whereas the incremental allows the whole system to be delivered as multiple installments added to the previous outcome. Each outcome is a result of rerunning the waterfall processes on a subset of the systems requirements. New functions are added to the existing system as constituents of the new subsystem [5,6].

Fig. 4 illustrates the procedures of developing an online shopping system using the waterfall and incremental models, respectively. It is assumed that a customer has four major actions with the system: Explore, Register, Order, and Pay. The business logic deals with the four corresponding actions as Searching, Verification, Invoice, and Transaction. The database maintains items of Catalogue, Customer, Stock, and Transaction Record. The payment transaction is linked to the Banking System outside the shopping system, which sends a copy of a transaction to the shopping system's database as a readable record. The waterfall model should deliver the whole system once after successfully completing the five stages in the end of the project.



(a)



(b)

Fig. 4. Developing an online shopping system using the waterfall model (a) and incremental model (b)

Following the incremental model, this system can be fully realized by two increments in succession. The first increment focuses on constructing a system enabling all 'internal' functionalities of the shopping system (Fig. 4b). This increment leaves the

procedure of Payment–Transaction–Banking aside because it requires interactions with a third-party outside the shopping system. When the first increment is completed, customers can do online shopping on this partial system. The payment can be made face-to-face by cash or cheque when the goods are presented to the customers. The next increment then adds the functionality of Payment–Transaction–Banking to the existing system to complete the whole system. It should be noted that the two increments should go through the same processes defined in the waterfall model, and the second increment only *adds* additional functions to the outcome of the first increment without reworking it. Otherwise it incurs a significantly high cost in system development.

3 Adaptive Concurrent Subsystem-Component Development Model (CSCDM)

The example shown in Fig. 4b illustrates the development procedure using the incremental model for two cycles. Experiences in supervising students E-commerce projects indicate that clients are more likely to demand an earlier release of a simple but yet usable system that meets their urgent business needs, rather than a partial system that should support the easy inclusion of the outcomes of subsequent increments. Clients are even prepared to compromise on the higher budget and longer duration for the system development in order to realise their business ambitions. The earlier releases are based more on business-oriented functional components, rather than structural groups.

Another common feature to this kind of E-commerce system development is that clients still treat the development as one project and want the final system to be deployed by the due date as whole. As long as their business data can be migrated to the new system, clients don't mind whether the new release needs to replace the previous release completely.

To guide the development for such E-commence systems, the author proposes a new system development model shown in Fig. 5. Its framework is the waterfall model, but its five processes are split into two loosely linked streams, one following the traditional system approach and the other adopting component-based method. The system stream follows the five stages of the waterfall model for conventional system development. The component-based stream iterates over the Implementation, Testing, and Deployment stages concurrently with the whole system process. This local iteration, or local increment, uses prototyping approach to realise the design of a component group for a quick deployment.

The prototype can be fully or partly absorbed into the corresponding system group, and/or modified to fit into the system design according to the client's feedback from using the prototype. Also the next iteration may include the previous prototype to produce a more functional prototype for the new component group. The intermediate outcomes of the system processes should not be used to improve the performance of a prototype of a component group because the deployed prototype will soon be replaced by the next prototype or the final system.

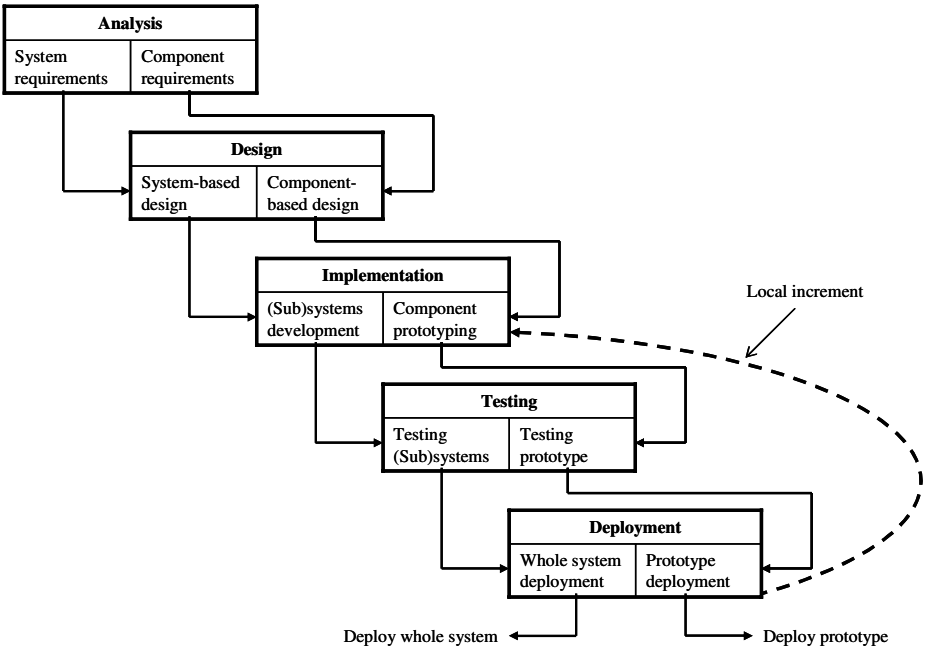


Fig. 5. The concurrent subsystem-component development model (CSCDM)

The key issues to the success of using this new model are to firstly confirm client's prioritization of business requirements, and secondly design the prototypes for the corresponding component groups. The first issue is to confine the potentially frequent changes in business requirements by the client during the quick prototyping because such changes will make the quick prototyping inefficient. Such changes may be accommodated in the system development stream. The design for the prototypes of different component groups should be carried out concurrently with the design for the whole system so that the prototypes can be absorbed into the system as much as possible. This implies that the design for all prototypes must be finalised in the end of the design stage so as to keep the highest possible consistency with the whole system design. This also explains why the prototyping iterates only over the Implementation, Testing, and Deployment stages.

4 Case Study: Client-Driven Online System Development for WCE Pizza

WCE Pizza is a recently opened business located near another established pizza shop. Its business operations and offerings are no difference from other pizza shops. Its business promotion strategy is to firstly attract the buyers who are the frequent customers of its next-door competitor by offering lower prices, and secondly retain its customer group by offering them a token-based rewarding scheme. This strategy has been very successful in its gaining of 45% pizza market there.

To make the business sustainable in the long term, WCE Pizza starts a new campaign to increase its customer base. The most important measure is to create a Web-based online system for attracting potential customers living in the radius of five or more suburbs from WCE Pizza. The current paper-based catalogue is effective within two surrounding suburbs. This online system will also make the business operation and management more efficient. The major requirements/tasks/outcomes for this online system are listed in Table 1 against the five stages of the waterfall model.

Table 1. Requirements, tasks and outcomes of the online business system for WCE Pizza

Analysis: Define system requirements	• to explore menu and special offers • to provide information for ordering • to allow customers register online • to give customers choices in making payment • to keep customer's service records • to link sales with stocks • to use a bulletin board for promoting special offers and events • to alert the registered customers by their emails any new special offers and events • to inform customers having sufficient rewarding points for free offers by their emails • to provide the system admin with tools for managing the system
Design: Specify overall and sub systems	• to develop a three-tiered client-server system • to create a series of user interfaces via Web browsers as the front system • to set up an Apache Web server and email server coexisting in the same computer to support business interactions using PHP and other means as the processing system • to set up a MySQL database in a separate computer as the data management system
Implementation: Develop subsystems leading to the integration of the overall system	• to develop the front system • to develop the processing system • to develop the database system • to articulate with the outside online payment system • to integrate all subsystems together
Testing: Test subsystems and the overall system	• to test individual subsystems • to test the integrated overall system
Deployment: Install and configure the system	• to install and configure the whole system • to migrate existing business record to the new system • to provide training and system maintenance in the agreed period

A three-tiered client-server system is proposed to be the best option for this system. Customers will use the Web browsers to interact with the system; an Apache Web server containing the Web documents and utilities and business applications along with an email server is hosted in the same computer as the middle processing layer; a MySQL database is used to store all business related data. The interactions among the subsystems will be enabled using PHP and other means. The final system should be delivered as whole in the end of the 22-week project.

Table 2. Development processes for WCE Pizza online business system using CSCDM

Instalment	Task	Outcome	Delivery
Prototype 1	• enable customers to explore menu and special offers through WEC Pizza Website; • provide customers with information for making order by phone.	Set up a simple Website for WCE Pizza.	Week 4
Prototype 2	• allow customers to register and make order online but to pay goods and/or services face-to-face; • keep customer's service records • link customers reward points with their purchases.	Set up database server and make it interact with the Web server without online payment facility.	Week 10
Prototype 3	• use a bulletin board for promoting special offers and events.	Set up a bulletin board for dynamic messaging.	Week 12
Prototype 4	• alert the registered customers by their emails any new special offers and events; • inform the registered customers who have accumulated sufficient rewarding point for free offers by their emails.	Set up the email server and customers' mailing list linked with their record in the database.	Week 15
Prototype 5	• link sales with stocks • provide the system admin tools for system management.	Add functions to both Web and database servers for tracking stock automatically; Add system management tools.	Week 20
Prototype 6	• make online payment available.	Make sure transactions through Web server, outside banking system, and database system maintain correct states.	Week 22
Final System Deployment	• install and configure the system • transfer existing business record to the new system.	Set up the whole system; Make sure the system fully functional.	Week 25

However, WEC Pizza cannot wait for five months for the whole system to be operational. A priority list is presented by WEC Pizza based on its business needs. After intensive negotiations, a mutual agreement based on seven instalments is reached between WEC Pizza and the project team. The seven instalments consist of six prototypes and a final system deployment. Tasks and deliveries of the seven instalments are listed in Table 2. Since the prototyping and its intermediate deployment consume extra time and resources, WEC Pizza is willing to extend the project from 22 weeks to 25 weeks and to compensate the additional costs. This client-driven E-commerce system was successfully realised by following the procedure of CSCDM shown in Fig. 5.

5 Conclusion

The concurrent system development model (CSCDM) is an alternative approach for guiding system development, especially for client-driven business systems. Its waterfall-based framework is easy to follow and understand by the developers. Its component-based stream only iterates over the Implementation, Testing, and Deployment stages, unlike the incremental model that iterates over all five stages. Since the design for the overall system and prototypes is done at the same stage, all the prototypes can be fully or partly absorbed into the corresponding system groups, and/or modified to fit into the system design. This makes the concurrent system development more efficient. The local iteration, or local increment, realises the business needs separately using prototyping without interference with the progression of the overall system development, which ensures the quick deployment of a prototype.

To make sure the success of using this new model, a client's business priority on requirements must be confirmed and the design for the prototypes of different component groups must be carried out concurrently with the design for the whole system so as to keep the highest possible consistency with the whole system.

However, the prototyping and its intermediate deployment require extra time and resources, so this model will not save the development costs, which must be made aware to the clients in the first instance.

References

1. Goldman, J.E., Rawles, P.T., Mariga, J.R.: Client/server information systems: a business-oriented approach. Wiley, Chichester (1999)
2. Ince, D.: Developing distributed and E-commerce applications. Addison-Wesley, Reading (2004)
3. Tanenbaum, A., van Steen, M.: Distributed systems: principles and paradigms. Prentice Hall, Englewood Cliffs (2002)
4. Coulouris, G., Dollimore, J., Kindberg, T.: Distributed systems: concepts and design. Addison-Wesley, Reading (2001)
5. Sommerville, I.: Software engineering. Addison-Wesley, Reading (2001)
6. Futrell, R.T., Shafer, D.F., Shafer, L.I.: Quality software project management. Prentice Hall, Englewood Cliffs (2002)

7. McManus, J., Wood-Harper, T.: Information systems project management. Prentice Hall, Englewood Cliffs (2003)
8. Royce, W.W.: Managing the development of large software systems. In: IEEE Proceedings WESCON, pp. 1–9 (1970)
9. Parnas, D.: Designing software for ease of extension and contraction, IEEE Transactions on Software Engineering, 128–138 (1979)

A Quantitative Approach for Ranking Change Risk of Component-Based Software

Chengying Mao

[1] School of Software, Jiangxi University of Finance and Economics,
330013 Nanchang, China
[2] School of Management, Huazhong University of Science and Technology,
430074 Wuhan, China
maochy@yeah.net

Abstract. The rapid evolution of component-based software brings great challenges to its maintenance in the later phase, so it is quite necessary to measure the risk its changes bring to the whole system. Through redefining the component dependency graph, the paper presents a two-step approach to assess the change risk of component-based software (CBS), which results from the partial changes of components in system. After obtaining the change risk of single component, we transform the component dependency graph to a component dependency tree, and then calculate the change risk of the whole CBS according to paths in the tree. In addition, a case study is made to illustrate how the technique works.

Keywords: Component-based software, change risk, component dependency graph, component dependency tree, slicing.

1 Introduction

During the past decade, component-based development has become an increasingly important means of constructing software systems [1], as well as one of the most popular forms of program composition. Despite this, the rapid evolution of component-based software brings great challenges to its maintenance in the later phase. Performing effective tracing, control and measurement to the system evolution is a hot topic in the direction of component-based software engineering (CBSE for short).

In general, component-based software (CBS) is built through the ways of composition and integration, and component is a "plug-in and play" encapsulated entity in software system. In this case, although it is quite favorable for component reuse, product upgrade and so forth, it will result in higher evolution speed than traditional software system. Although it is easy to replace a component syntactically, making sure that the modified system still works is a challenge. In order to facilitate the maintenance activities and their optimum configuration, it is necessary to measure the changes. Most existing researches pay much attention to the reliability of software system [2],[3],[4],[5]. However, from the perspective of system maintenance, the maintainers mainly concern on the question that how much risk exists in the software,

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 92 – 103, 2007.
© Springer-Verlag Berlin Heidelberg 2007

because the risk directly means the amount of effort which they should spend during the process of maintenance. For example, the risk-based testing strategy [6] is just based on this idea.

Risk assessment is an essential part in managing software development and maintenance. Risk is usually defined as the probability of an error occurring (chance of failure) related to the damage expected when this error does occur [6]. In this paper, we merely discuss the risk of CBS which is caused by component change. The change risk of CBS can help maintainer to allocate resource of time and man-power in the later stage of software life-cycle. For instance, the higher change risk means the higher cost of the regression testing of CBS. Therefore, it's not hard to find that change risk analysis is a useful activity in the later maintenance of component-based software development.

The remainder of this paper is organized as follows. In Section 2, we briefly survey the related work in this research direction. The denotation of CBS and the concept of component change are described in Section 3. In Section 4, we discuss the analysis approach for CBS's change risk in details. We also illustrate the value of the approach using a case study in Section 5, and Section 6 concludes the paper.

2 Related Work

Here, we give a brief survey on the works that are correlative to ours. At the early research stage, most investigators concern on the reliability evaluation of CBSs. The typical works are as follows: Through using a discrete time Markov process to model the transfers of control between components, then estimate the reliability of CBS according to system's execution profile. Besides, the testing resource allocation problems are also be solved in the light of this model [2],[3]. Based on function abstractions, Mao et al. [4] proposed a general model for component-based software reliability, i.e., *component probability transition diagram*. The model focuses on reliability tracing through the dynamic development process. Moreover, Gokhale et al. developed an approach to assess the reliability of an application by taking into consideration component-level fault tolerance [5]. Differing from the above investigation, we don't consider how much ability a CBS has achieved, but devote our minds to its risk caused from component change.

Software change is an essential operation for software evolution. Change impact analysis provides techniques to address the problem by identifying the likely ripple effect of software changes and using this information to re-test the software system, i.e., regression testing. Zhao et al. [7] employed *slicing* and *chopping* techniques in the component-based software, and assessed the effect of changes in a CBS by analyzing its formal architecture specification (*architecture description languages*, ADLs). Their change impact analysis technique is applied at the architecture level to support architecture evolution during software design. However, the change impact analysis is mainly used at the code level of CBS in this paper.

While considering the risk analysis approach for component-based software, Yacoub et al. described a heuristic reliability risk assessment method based on scenarios analysis and dynamic metrics [8], which can also be used to identify critical components in CBS. On the other hand, Goseva-Popstojanova et al. [9] used the UML and

commercial modeling environment RoseRT to obtain UML model statistics. After a Markov model is constructed to obtain scenarios risk factors, the overall system risk factor are estimated using them. The above two approaches are both designed to evaluate the reliability risk at the early stage of software life cycle. However, we perform change risk analysis while one system version evolves to another version. At that time, the program code is implemented. In addition, the former two approaches aim at a specific software version, and don't consider the evolution characters of system risk under the direction of time sequence. So the paper focuses on the risk of system change in comparison with the old version.

3 CBS and Its Changes

3.1 Description Model for CBS

After analyzing researchers' broad agreement on the meaning of the term "*component*" [10], we define it as a nearly independent, replaceable module, which encapsulates data and operations used to implement some specific functions. Component can be constructed by many programming technologies. Generally, Object-Oriented programming (OOP) provides the basic technique support to the construction of component, so most components are developed by the Object-Oriented programming languages at present, such as Java, C++ and C#.

Component-based software is composed of some components (they are maybe heterogeneous) through the connectors, which is a loosely-coupled structure. Generally speaking, the potential failure risk of the whole component-based system greatly relies on the failure risks of single component or connector. In this paper, we mainly discuss the fallibility risk of the whole software system resulting from the changes in one or more components. During employing architecture analysis, testing or reliability evaluation to a CBS, several kinds of denotations are proposed to model it, such as the following examples. Reference [11] combines the formally-extended UML and *labeled transition systems* (LTS for short) to describe the system. Wu et al. utilize the *component interaction graph* (i.e., CIG) to model the interaction scenarios between components. Moreover, the representations such as dependency matrix [12], component probability transition diagram [4] are also introduced to represent the CBS. However, the most popularly accepted model is still the component dependency graph (CDG)[2],[3],[8],[9]. Here, we redefine the CDG so as to analyze software's change risk.

Definition 1 (*Component Dependency Graph*, CDG)**.** A component-based software which can be denoted by the dependency graph as a 4-tuple $CDG = (C, T, n_s, n_t)$, where C represents the node set of components, T is the set of edges used to represent the connectors, n_s and n_t are the start node and termination node respectively. The node set of components and edge set of connectors can be further defined as follows.

$C = \{< c_i, cpx_i, rf_i >\} (1 \le i \le |C|)$, where c_i is the identifier of the ith component, cpx_i is the complexity of component c_i, and rf_i is the failure risk of component c_i (i.e., the severity degree of the results caused by the failure of component c_i).

$T = \{< t_{ij}, p_{ij} >\}(1 \le i, j \le |C|)$, where t_{ij} is the identifier of the connector from component c_i to c_j, here c_i is called source component and c_j called target component. p_{ij} is the probability of the connector t_{ij} being executed. In general, the following formula can be given, where k is sum of all out edges of component c_i.

$$\sum_{j=1}^{k} p_{ij} = 1 \tag{1}$$

In the rest of this subsection, the factors of cpx_i, rf_i and p_{ij} will be addressed in details.

3.1.1 Complexity of Component

As viewed from the visibility of component code, components can be classified into in-house (or inner) component and external component. The code of inner component is usually visible to developers (or maintainers) of CBS, whereas the code of external component (such as third-party COTS) is unavailable in most situations. Here, we discuss the complexity of a component in two cases, as follows.

When the code of a component is available, we think that the complexity of program's control flow graph (CFG), line number of codes and the number of included library files are three main factors which can affect the complexity of the whole component. For component $c_i (1 \le i \le |C|)$, its complexity (i.e., cpx_i) can be calculated as follows.

$$cpx_i = w1 \cdot mcc(c_i) + w2 \cdot hal(c_i) + w3 \cdot lib(c_i) \tag{2}$$

Where $w1$, $w2$ and $w3$ are the adjustment weight, and here $w1 = 0.6$, $w2 = 0.3$ and $w3 = 0.1$. $mcc(c_i)$ is the relative McCabe complexity of component c_i, $hal(c_i)$ is the relative Halstead complexity of component c_i, and $lib(c_i)$ is the relative library inclusion complexity of component c_i. Here, the calculation method of $lib(c_i)$ is provided in formula (3), other two factors can be measured in the similar way.

$$lib(c_i) = \frac{incld(c_i)}{\max\limits_{1 \le j \le |C|} \{incld(c_j)\}} \tag{3}$$

Where $incld(c_i)$ represents the number of library files included in the program of c_i.

In the other case, i.e., component's code is invisible, component providers generally provide skeleton information (also called *metadata*) of their components to users through the standard exchangeable format such as XML files. The typical information includes total lines of code, methods number of method call relation in component, etc. We can roughly measure the complexity of the component using the skeleton information. For example, the methods and method call relation in component c_i can be modeled via the *method call graph* (MCG) which is defined in our previous work (as for detailed MCG, please refer to [13]). Then, the approximate complexity of component c_i can be calculated by the following equation.

$$cpx_i = w1 \cdot \frac{l(c_i)}{\max\limits_{1 \le j \le |C|} \{l(c_j)\}} + w2 \cdot \frac{|V_{MCG(c_i)}| + |E_{MCG(c_i)}|}{\max\limits_{1 \le j \le |C|} \{|V_{MCG(c_j)}| + |E_{MCG(c_j)}|\}} \tag{4}$$

Where $l(c_i)$ represents the total line number of component c_i, $|V_{MCG(c_i)}|$ and $|E_{MCG(c_i)}|$ are the method number and theirs call times in the MCG of component c_i, respectively. $w1$ and $w2$ are the weight coefficients, here $w1 = w2 = 0.5$.

3.1.2 Failure Risk of Component

To measure the failure risk of a component is actually a task of analyzing the severity degree of the results once that component can't be executed in the normal course. Here, we also consider this problem in two cases: code is visible or not.

For visible-code component, evaluator can adopt the mutant analysis technology [14] to achieve this task: First, seed some typical faults into the program of the component. Then determine the risk coefficient by executing the whole system and observe its failure behaviors. In this paper, we adopt MIL_STD 1629A Failure Mode and Effect Analysis [15] to assign the failure risk of each component. According to that standard, the value of failure risk ranges from 0.25 to 0.95, that is, 0.25, 0.5, 0.75 and 0.95.

For invisible-code component, failure risk assessment can be performed along the following two ways: (1) Organize a few experienced software designers to review the specification documents of component, then rank the failure risk of that component using the method of *fuzzy comprehensive evaluation*. (2) Generally speaking, if a component lies on the joint of information exchange flow in CBS, its failure will lead to more serious results, otherwise not. Hence we can calculate the failure risk of a component according to its fan-in or fan-out in the CDG of system. Moreover, the *PageRank* algorithm [16] in the field of information retrieve can also be introduced to rank the importance of a component in CBS, so as to compute its failure risk. Whichever way is adopted, the higher weightiness of a component means higher failure risk if the component fails.

3.1.3 Execution Probability of Connector

Given a connector $t_{ij}(1 \le i, j \le |C|)$, its execution probability is essentially the transition probability from component c_i to component c_j. Generally, the probability is statistically computed on the basis of all possible execution scenarios of the component-based software. Here, the method mentioned in Reference [3] is adopted to calculate the probability.

$$p_{ij} = \sum_{k=1}^{|S|} P_{s_k} \cdot \left[\frac{|Interact(c_i, c_j)|}{|Interact(c_i, c_l)|_{1 \le l \le |C| \wedge l \ne i}} \right] (c_i, c_j, c_l \text{ in } s_k) \tag{5}$$

Where S represents the set of system scenarios, P_{s_k} is the execution probability of the scenario s_k, and $|Interact(c_i, c_j)|$ represents the interaction times from component c_i to c_j in a specific scenario, such as s_k. It should be noticed that the value of $|Interact(c_i, c_j)|$ generally doesn't equal to that of $|Interact(c_j, c_i)|$.

3.2 Changes in CBS

During the evolution process of component-based software, the system should be modified for purposes of correcting errors, improving its performance or enhancing

some function points. In common situations, the modifications of CBS can be summarized in the following three aspects: (1) Fractional code in component is modified, (2) component is added into CBS, or (3) component is deleted from CBS. Among them, the first one is the most prevailing fashion, so we merely take it into account in current research.

To ease our discussion, we define two concepts about change as follows.

Definition 2 (*Changed Statement Set*)**.** Given a component c, the set of the modified statements (including added and deleted statements) in it is called changed statement set (CSS), and denoted as $S_{stm}(c) = \{s_1, s_2, \cdots, s_m\}$, where m is total number of changed statements in component c.

Definition 3 (*Changed Method Set*)**.** Given a component c, the set of the modified methods (including added and deleted methods) in it is called changed method set (CMS), and denoted as $S_{mth}(c) = \{m_1, m_2, \cdots, m_n\}$, where n is total number of changed methods in component c.

4 Change Risk Analysis

While maintainer modifies some components in order to satisfy new introduced requirements or ensure requirements to be correctly implemented, it is very necessary to consider the questions such as which parts will be affected by the changes, or how much risk it will bring to the whole system, etc. The change risk analysis proposed in this paper is carried out after change and before regression testing, and is used to evaluate the probability of additional fallibility caused by component's changes. The evaluation result will direct the deployment and implementation of subsequent regression testing activities, such as plans for testing fund and schedule, and test case selection.

Change risk analysis can be performed in two steps: (1) For each modified component, analyze how much influence its changes brings to itself. (2) Based on the analysis for single component, calculate change risk of the whole system.

4.1 Change Risk of Single Component

Before performing change risk for a component, it is necessary to calculate the proportion of the changed parts to whole component. In general, the higher proportion means the greater potential risk, and vice versa.

Case 1: The code of component is visible. According to changed statement set, the affected statements can be computed via the static slicing technique [17], and then the change ratio is also obtained. At fist, we give the definitions about definition variable and use variable as follows.

Definition 4 (*Definition/Use Variable*)**.** For any statement s, if the variable v in statement s to be assigned a new value, we call v as a definition variable of s, and denote it as $def(s,v)$. Similarly, if the statement s references the value of v, we call v as a use variable of s, i.e., $use(s,v)$.

Given a changed component c, its change ratio $ratio(c)$ can be calculated using the following algorithm, where, $forward_slice(s_i, v_j)$ is the forward static slicing based on slice criterion (s_i, v_j), similarly, $backward_slice(s_i, v_k)$ is the backward slicing according to criterion (s_i, v_k).

Algorithm 1. Calculate $ratio(c)$

Input: $S(c)$ is the statement set of component c;
$S_{stm}(c)$ is the changes statement set of component c.
Output: $ratio(c)$ is the change ratio of component c.

```
{
  S_tmp = ∅ ; //temporary statement set
  for each s_i ∈ S_stm(c)
  {
      for each def(s_i, v_j) in s_i
        S_tmp = S_tmp ∪ forward_slice(s_i, v_j) ;
      for each use(s_i, v_k) in s_i
        S_tmp = S_tmp ∪ backward_slice(s_i, v_k) ;
    }
    return |S_tmp| / |S| ;
}
```

Case 2: The code of component is invisible. In this case, it needs to construct method call graph MCG [13] according to the metadata information provided by component developers. For each $m_i \in S_{stm}(c)(1 \le i \le n)$, the method set (denoted as $affected(m_i)$) affected by m_i can be found out through forward and backward traversal starting from m_i in MCG. Suppose the set of methods in MCG is M, the change ratio of component c can be calculated as below,

$$ratio(c) = \left| \bigcup_{i=1}^{n} affected(m_i) \right| \Big/ |M| \tag{6}$$

Finally, for each $c_i \in C$, its change risk cr_i can be computed via formula (7). Obviously, the change risk of the unchanged component is 0.

$$cr_i = ratio(c_i) \cdot cpx_i \cdot rf_i \tag{7}$$

4.2 Change Risk of CBS

In fact, change risk measurement for the whole CBS is to analyze the reliability impact caused by the modified components. To facilitate the risk analysis, we transform CDG into a spanning tree, called component dependency tree (CDT).

Definition 5 (*Component Dependency Tree*)**.** For a $CDG = (C, T, n_s, n_t)$, it can be transformed into a spanning tree: $CDT = (N, E)$, where $N = \{< c_i, cr_i >\}$ is the set of

component nodes, and $E = \{< e_{ij}, p_{ij} >\}$ is the set of directed edges which represent component execution dependency. Obviously, $1 \le i, j \le |C|$.

A CDG can be transformed into a CDT in the following steps:

Step 1: Select the initial node n_s as the root node of CDT, and label it as current node.

Step 2: For any current node, perform *breadth first search* (BFS) to the CDG, treat each edge of current node in CDG as an edge of the corresponding node in CDT, and regard the target node of this edge as the child node of current node. The node in CDT will terminate its expansion while it satisfies either one of the following two conditions: (1) If the current node is the termination node n_t, or (2) the current node (assume it lies in *i*th layer in tree) has appeared in the lower (i.e., from 1 to *i*-1) layers in the tree.

Step 3: Carry out the second step iteratively until all nodes in the tree can't be expanded.

In component-based software system, components interact on each other through modifying the execution information flow. In general situation, some component's modification will affect the information flow in its execution scenarios, and thus brings side effect to the followed components. Even if the changed component doesn't fail in some scenario, it is still possible to cause other unmodified components to fail. Therefore, it is very necessary to assess the potential impact resulting from the changed components.

Based on the CDT, the risk of changed component and its propagation can be well assigned via algorithm 2, whose key step is to determine the *propagation factor* θ_p. We primarily choose the formula $\theta_p = 1/(1+\alpha h)$ for computing, where h is the difference of layer between affected component and changed component, and α is the adjustment coefficient.

Algorithm 2. Assign cr

Input: CDT is component dependency tree of CBS;
C is the component set of system;
C_{chg} is the subset of changed components.
Output: CDT' is the new CDT expanded with change risk.

```
{
 for each node c_k in CDT
   cr_k = 0 ;
 for each c_i ∈ C_chg
     fill change risk to the corresponding node of c_i in CDT ;
  for each c_i ∈ C_chg in CDT
    for each offspring c_j of c_i in CDT
    {
       calculate the layer difference between c_i and c_j , denoted
         as h_ij ;
```

$cr_{new} = ratio(c_i) \cdot cpx_i \cdot cpx_j \cdot rf_j / (1 + \alpha h_{ij})$;
if($cr_{new} > cr_j$)
$cr_j = cr_{new}$;
}
return new CDT(i.e., CDT') after assigning change risk;
}

It is worth noting that the change risk of start node n_s and termination node n_t are all assigned to 0. After getting the complete CDT, the change risk of whole CBS (denoted as CR) can be calculated by algorithm 3. Here, the path from root node n_s to each leaf node c_k in CDT is denoted as $path =< n_s, \cdots, c_i, c_j, \cdots c_k >$.

Algorithm 3. Calculate CR

Input: CDT is component dependency tree of CBS.
Output: CR is the change risk of CBS.

{
$R_{sum} = 0$;
for each leaf-node c_k in CDT
{
construct a path from root to c_k , i.e.,
$path =< n_s, \cdots, c_i, c_j, \cdots c_k >$;
$R_{tmp} = 1$;
for each node c_i in $path$ and c_j is the next node of it
$R_{tmp} = R_{tmp} \cdot (1 - cr_i) \cdot p_{ij}$; //when calculating to c_k , $p_{k*} = 1$
$R_{sum} = R_{sum} + R_{tmp}$;
}
$CR = 1 - R_{sum}$;
return CR ;
}

5 A Case Study

In order to describe the above calculation methods more clearly and validate their effectiveness, we choose a simple CBS as a case to study. The architecture of the sample system is adopted from Reference [3],[8]. Partial information such as component's complexity and failure results is directly designated due to unavailability of more detailed information about components in that system. That is, the detailed computing procedure shown in Section 3.1 is omitted here for the sake of conciseness. Finally, the component dependency graph of the sample CBS is shown in Figure 1.

According to the transformation method, the component dependency graph in Figure 1 can be converted into a component dependency demonstrated in Figure 2.

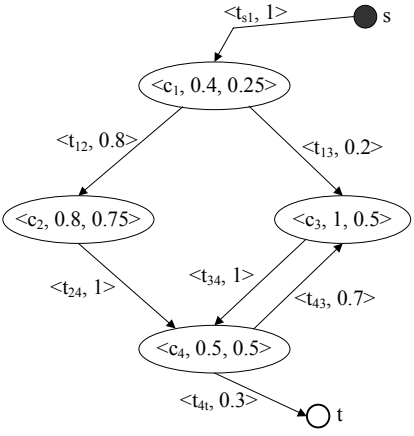


Fig. 1. The component dependency graph (CDG) for a sample component-based software system

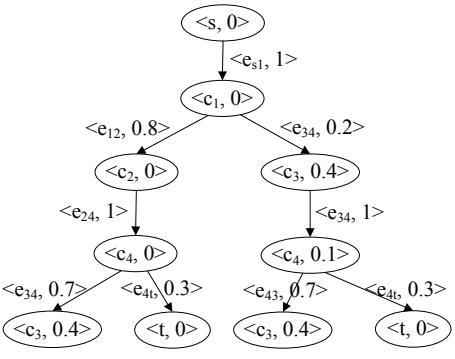


Fig. 2. The corresponding component dependency tree (CDT) of the above CDG

Suppose component c_3 is changed and its change ratio $ratio(c_3) = 0.8$, then the change risk of c_3 is $cr_3 = 0.8 \times 1 \times 0.5 = 0.4$. The propagation effect of that changed component in CDT can be calculated and is shown in Figure 2: The affected component of c_3 is component c_4, $\theta_p = 1/(1+1\times1) = 0.5$, where $h_{34} = 1$ and α is assigned 1. Therefore, $cr_4 = 0.8 \times 1 \times 0.5 \times 0.5 \times 0.5 = 0.1$. Apart from component c_3 and c_4, other components' change risks are all zero.

Then, for the path in CDT $path_1 = < s, c_1, c_2, c_4, c_3 >$, $R_{mp} = 1\times1\times1\times0.8\times1\times1\times1\times0.7\times 0.6 = 0.336$. After computing other three paths in the same way, the change risk of whole CBS can be obtained as below, $CR = 1 - (0.336 + 0.24 + 0.0454 + 0.0324) = 0.3462$.

Except the case of only one changed component in CBS, several changed components may exist in the system. For example, if component c_2 and c_3 are both changed ($ratio(c_2) = 0.4$ and $ratio(c_3) = 0.8$), the change risk of whole system is $CR = 1 - (0.2451 + 0.1751 + 0.0454 + 0.0324) = 0.502$.

Furthermore, more profound analysis can be carried out in this case, and it is not hard to draw some conclusions about change risk. (1) $0 \le CR \le 1$. (2)From the view of execution scenarios, the change in frequently-executed component brings more threat to system's reliability. (3)The higher change ratio means the higher change risk. (4)The change risk of whole system is also positively related to complexity and failure result of component.

6 Concluding Remarks

With the great advancement of component-based software technology, it has been increasingly adopted in the development of large-scale complex software systems. However, the problems about evolution process control and change risk analysis haven't been settled perfectly and are still the open issues in CBSE. Through redefining the component dependency graph, the paper presents a novel approach to assess the change risk of component-based software, which is caused from the partial changes of components in system. Based on obtaining the change risk of single component, we transform the component dependency graph to a component dependency tree, and then calculate the change risk of the whole CBS according to paths in the tree. In addition, a case study is also made to illustrate how the technique works.

Currently, of course, our research work is early——that is, there are a few issues that need to explore further, such as sensitivity analysis and further experiments validation.

Acknowledgments. Thanks to the anonymous reviewers for helpful suggestions on early version of this paper. This work was supported in part by the National Natural Science Foundation of China under Grant No.70571025, the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No.20060487005, the Natural Science Foundation of Hubei Province of China under Grant No.2005ABA266, the Science Foundation of Education Bureau of Jiangxi Province of China under Grant No.GJJZ-2007-267, and the Young Scientist Foundation of Jiangxi University of Finance and Economics.

References

1. Pour, G.: Component-Based Software Development Approach: New Opportunities and Challenges. In: Proc. of Technology of Object-Oriented Languages, pp. 375–383 (1998)
2. Lo, J.H., Kuo, S.Y., Lyu, M.R., Huang, C.-Y.: Optimal Resource Allocation and Reliability Analysis for Component-Based Software Applications. In: Proc. of COMPSAC'02, pp. 7–12. IEEE Press, New York (2002)
3. Yacoub, S., Cukic, B., Ammar, H.H.: A Scenario-Based Reliability Analysis Approach for Component-Based Software. IEEE Transaction on Reliability 53(4), 465–480 (2004)
4. Mao, X.G., Deng, Y.J.: A General Model for Component-Based Software Reliability. Journal of Software 15(1), 27–32 (2004)
5. Gokhale, S.S.: Software Reliability Analysis with Component-Level Fault Tolerance. In: Proc. of Annual Reliability and Maintainability Symposium, pp. 610–614. IEEE Press, New York (2005)

6. Ottevanger, I.: A Risk-Based Test Strategy. IQUIP Informatica B. V, pp. 1–13 (1999)
7. Zhao, J., Yang, H., Xiang, L., Xu, B.: Change Impact Analysis to Support Architecture Evolution. Journal of Software Maintenance and Evolution: Research and Practice 14, 317–333 (2002)
8. Yacoub, S., Ammar, H.H.: A Methodology for Architectural-Level Reliability Risk Analysis. IEEE Transaction on Software Engineering 28(6), 529–547 (2002)
9. Goseva-Popstojanova, K., Hassan, A., Guedem, A., Abdelmoez, W., Nassar, D.E.M., Ammar, H., Mili, A.: Architectural-Level Risk Analysis using UML. IEEE Transaction on Software Engineering 29(10), 946–959 (2003)
10. Wu, Y., Pan, D., Chen, M.H.: Techniques for Testing Component-Based Software. In: Proc. of ICECCS'01, pp. 222–232. IEEE Press, New York (2001)
11. Liu, W., Dasiewicz, P.: Component Interaction Testing using Model-Checking. In: Proc. of Canadian Conference on Electrical and Computer Engineering, pp. 41–46. IEEE Press, New York (2001)
12. Li, B., Zhou, Y., Wang, Y., Mo, J.: Matrix-Based Component Dependency Representation and Its applications in Software Quality Assurance. In: ACM SIGPLAN Notices, vol. 40(11), pp. 29–36. ACM Press, New York (2005)
13. Chengying, M., Yansheng, L.: Regression Testing for Component-based Software via Built-in Test Design. In: Proc. of the 22nd Annual ACM Symposium on Applied Computing (SAC'07), ACM Press, New York (to appear, 2007)
14. Acree A.T., Budd T.A., DeMillo R.A., Lipton R.J., Sayward F.G.: Mutation Analysis. Technical Report GIT-ICS-79/08, School of Information and Computer Science, Georgia Institute of Technology (1979)
15. Procedures for Performing Failure Mode Effects and Criticality Analysis. US MIL_STD_1629A/Notice 2 (1984)
16. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Technical Report, Computer Science Department, Stanford University (1999)
17. Tip, F.: A Survey of Program Slicing Techniques. Programming Languages 3(3), 121–189 (1995)

Relating Software Architecture Views by Using MDA

Rogelio Limon Cordero and Isidro Ramos Salavert

Department of Information Systems and Computation, Technical University of Valencia,
Camino de Vera s/n E-46022 Valencia – Spain
{rlimon, iramos}@dsic.upv.es

Abstract. Software architecture views separate the concerns that are presented in phases previous to the detailed design phases of a software system. Each view represents a different structure of a system for its analysis and documentation. Although each view can be designed separately, they are linked because they are part of the same system. These links must be explicitly shaped and maintained in order to achieve synergy in their design and also to preserve consistency among the views when they change for evolution or maintenance reasons. This work proposes the use of the Model-Driven Architecture (MDA) approach to establish the links among the architectural views through a relation of models. This work shows how relations among software architectural views can be established by means of strategies used in MDA in a systematic way.

Keywords: Software Architecture, Modeling Driven Architecture.

1 Introduction

Nowadays, software architecture (SA) is used in the development of complex software systems to shape the fundamental structures that support the detailed design phase of these systems. Software architectural views (SAV) represent the different structures that a system can have and shape the concerns present at the design level in different ways.

Even though architectural views can be shaped separately, these views establish relations among each other that help maintain the cohesion of the system. These relations can be traced and their corresponding elements at lower levels of design can also be traced.

The relations of each architectural view must be identified and explicitly shaped in order to achieve the following objectives: i) generate a view using another view as reference, ii) make rules to assure conformance between the architectural view and the code, iii) preserve the consistency among views when a view is modified (for maintenance or evolution), and iv) document the software architecture.

The studies made to date do not fully cover all the objectives stated above. For objective (i), there are methods such as the Attribute-Driven Design (ADD) approach presented in [1], which uses a view to design another view. However, the relations between them not are explicitly specified and are not recorded. For objective (ii) there are many proposed methods; however their solutions as demonstrated are incomplete as stated in [2]. For objective (iii), a specific algebra to deal with the consistency is

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 104 – 114, 2007.
© Springer-Verlag Berlin Heidelberg 2007

used in [3], but it only focuses on one type of view. [4] deals with relations of multiple views in a formal way, but its concept of view is too specific to apply to other types of views. For objective (iv), there are many studies for documentation, e.g. [5].

The main goal of this work is to use a Model-Driven Architecture (MDA) proposal to deal with the view relations and to achieve the objectives described above. This work presents a strategy to relate two SAVs and generate one SAV from another one, by means of relations and the transformation of models.

This paper is structured as follows: Section 2 shows the relation between SAV and MDA. It explains how to use the MDA to relate SAVs with each other by means of a transformation process among the architectural models. Section 3 identifies the relations among the meta-models of the SAVs considered. Section 4 shows how to relate SAVs at the model level. Section 5 illustrates this approach in a case study, and section 6 presents our conclusions.

2 Relating SAVs and the Model-Driven Architecture

2.1 Software Architecture Views

The SA is shaped by a set of views that represent the different structures of a software system [1]. The views can be used to separate concerns and several models can be used to represent them.

There are several approaches that describe what the architectural view should be. Among the most relevant are the following: SEI [1], IEEE-1471 [6], and Kruchten [7]. Even though there are some differences in how they interpret a view, there are coincidences in the type of elements and relations that are considered in each one. In all of these approaches, components and connectors are considered as basic building blocks, and the module element is included in different ways.

The focus in this work is to analyze and deal with the relations between at least two types of architectural views. Therefore, the component-connector(C&C) and modular view-types proposed in [1] have been chosen as the most appropriate approaches to apply to our proposal.

The main goal is to shape the correspondence rules at the meta level and perform the transformation at the model level.

Figure 1(a) shows an example of two correspondence relations at the meta-model level: in this case, R1 indicates that each module element (Module) of the <<*Modular View*>> has a correspondence with an element (component) of the <<*C-C View*>>, and R2 indicates that a <<*Uses*>> of the <<*Modular View*>> has a correspondence

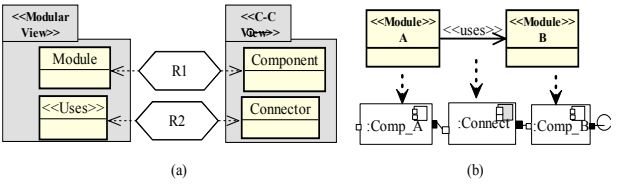


Fig. 1. Example of: (a) Relations between the two views at the Meta-model level, (b) Transformation from a view model to another view model

with a connector element of the <<*C-C View*>>. These relations are used to achieve the transformation at the model level as shown in Figure 1(b). This is an example of objective (i) of the relation of software architecture views, described in section 1.

2.2 The Model-Driven Architecture and the Software Architecture Views

The MDE is a novel paradigm that reduces the software complexity introduced by the current diversity in technology. It uses methodologies and support tools that are based on models with a high level of abstraction (meta-models) instead of those of a specific technology. MDA [8] is a proposal of OMG (Object Management Group) for MDE.

MDA creates system models with a high level of abstraction. These models can be refined using a series of transformations to obtain different models at the code level. [9]. The highest model is called PIM (Platform-Independent Model), and the model of a specific platform is called PSM (Platform-Specific Model). The PSMs are generated by a transformation from PIM to PSM.

SAVs can be shaped as PIMs, even though they represent specific systems. They have features that are applicable to a set of similar systems (families of software systems) so that the same SAV can be adapted to systems that have different platforms, i.e., to PSMs. Furthermore, SAVs are formed with elements of a high level of abstraction, which are shaped by means of languages (ADLs) or notations that are independent of the technological platform. These elements can be mapped at code level. Therefore, the MDA can be used not only to refine a SA, but also to make transformations among architecture views that are at the same level [9].

2.3 Using MDA to Build Relations Among SAVs

Three types of relations among the SAVs studied in this work have been identified: *i*) the relations that are present between the modular view and the C&C view. These generate a view at the model level using each other as reference; *ii*) the relations that are formed between the C&C view model and other models that define the interactions among the components (scenarios, maps of use cases).These other models are used as references to refine the C&C view model; and *iii*) the relations established when a view is adapted to a specific system (at different levels of abstraction).

The relations *i* and *ii* are shaped using the MDA frame as shown in Figure 2. The Meta Object Facility (MOF) [10] provides the language for expressing the models. It

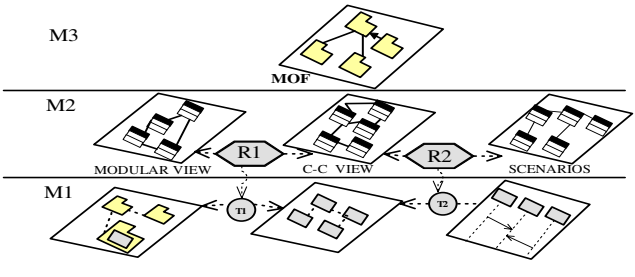


Fig. 2. Relations among meta-models and transformation among models

is used to specify itself (level M3) and to specify the meta-models of the views and their interactions, which are given by the scenarios meta-model (level M2). The relations R1, R2 and R3 are also established at level M2. R1 is related to *i*, and R2 is related to *ii*.

The next level (M1) is where the transformations T1 and T2 are carried out. A first version of the C-C view model is obtained by transformation T1, using the modular view model as source. This first version is refined by means of T2 using the scenarios model a refinement.

2.4 Transformation Process of the Architectural Views Models

In order to achieve the transformations among the view models, a process that allows the design of the C&C view using the modular view and the scenario model as inputs, is proposed.

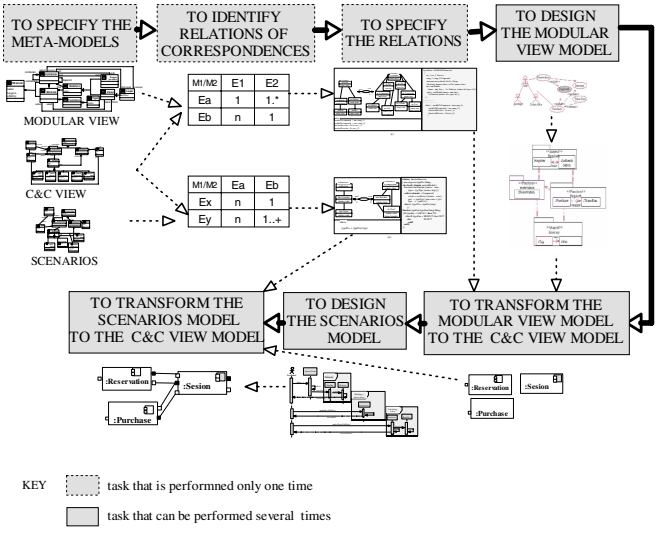


Fig. 3. The tasks and elements involved in the transformation process of the architectural view models

The tasks and elements involved in this transformation process are shown in Figure 3. The meta-model specifications of the modular view, the C&C view, and scenarios are the first elements that are required to start this process. Next the correspondence relations between the modular view and the C&C view, and between the scenarios and the C&C view are identified, and specified by mean of a language and/or notation. The QVT-Relational language and the MOF notation are used for this task. All these tasks are performed only one time since their products (elements) are used as a template for the transformation models.

The design task of the modular view model is the first task that can be performance several times as required. This task uses requirement as reference, these requirements can be expressed by use cases.

The modular view model and the relations established between the two meta-model views are used as source in the transformation task, which produces a first version of the C&C view model. Next, a second transformation is carried out to refine to the produced model. To do it the scenarios model and the first version of the C&C view model are used as source to produce a refined version of this view model.

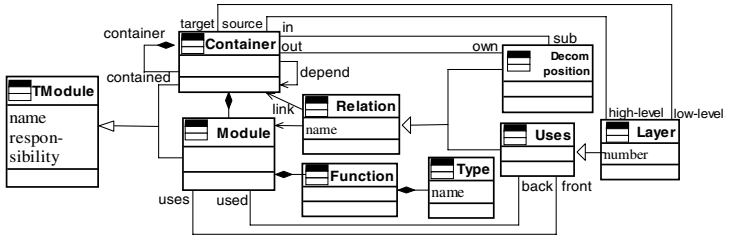
3 Specifying the Meta-models

For the conformation and specification of the meta-models of the modular and C&C views, several proposals [5][6][7] have been analyzed. The MOF graphic notation has been used for the specification of these meta-models.

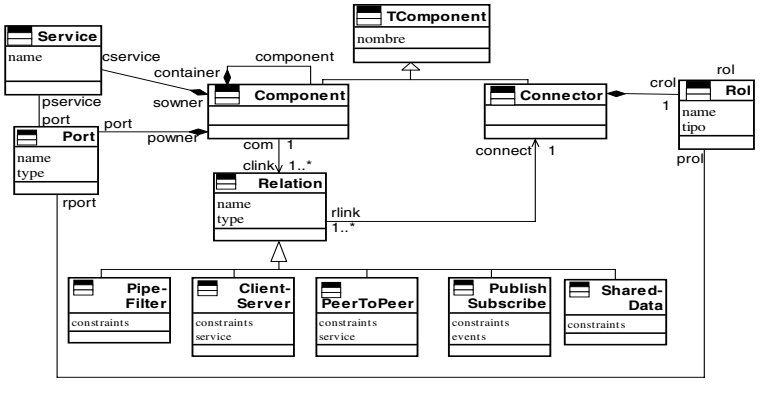
3.1 Specification of the Modular View Meta-model

The main element considered for this view is the module. We have made a distinction between a module that is used as a container for more modules and the module that only carries out a single function.

In Figure 4(a) shows the meta-model for this view. The relation class represents the styles presented in [5] (decomposition, uses and layers). These styles are treated as



(a)



(b)

Fig. 4. (a) Modular view meta-model, (b) C-C view meta-model. (MOF notation).

relations since what distinguishes one style from another is the type of relation. The labels in the links are useful for indicating how the relation is made; for example in Figure 4(a) the '*uses'* label indicates which element of the *Module* class will be used by another element of the *Module* class.

3.2 Specification of the C-C View Meta-model

The *C-C* view meta-model is depicted in Figure 4(b). This Figure shows the *Component* class and *Connector* class as the main elements. Both are derived from a more general component class (*TComponent*). Similarity to the above meta-model, the way in which these elements are related corresponds to their styles or relations (Pipe-Filter, Client-Server, PeerToPeer, Publish-Suscribe, Shared-Data). These styles are inherited from the *Relation* class. This class links the components by means of the *Connector* class.

The interactions among the components of this view are refined using the scenarios meta-model; this meta-model specifies how the components (objects) communicate (by means of events or messages). The scenario meta-model is specified by the UML 2.0 meta-model [11].

4 Establishing Relations Among the Meta-models

Before the relations are shaped, a language must be chosen to represent them correctly. For the specific case of MDA, OMG proposes the Meta Object Facility (MOF) to express the meta-models. It uses to Query/Views/Transformations (QVT) [10] to establish the relations among these meta-models. Specifically, the QVT-relational language is used to describe the relations.

4.1 Identifying Correspondence Relations

The correspondence among each element of the meta-models previously specified must be identified, taking into account the way in which their rules of relation are represented through QVT. The source and target meta-models must be identified first. In this case the source meta-model is the modular view. The target meta-model is the C&C view model, since its model is obtained from the model of the modular view. This means that the rules considered in the relations for the elements of the modular view can only be of type "**check only**", to verify the elements of the modular view, and the type "**enforce**", to create the elements of the C&C view.

Table 1. Relations identified in the modular and C&C meta-models

Relation	Type	Classes involved by view	
		Modular view	C-C view
moduleToComponet	top	Module	Component
functionToService	-	Module, Function, Type	Component, Service, Port
rUsoModToConnector	top	Uses, Module, Service	Connector, Role, Component, Port; Service
rCompoModToCom	top	Composition, module	Component

Note that not all the elements between the models considered have correspondence relations. Table 1shows some of the identified correspondence relations, their name, type (according QVT), and the elements involved (classes).

4.2 Specification of the Relations Between the Software Views

The relations shown in Table 1 shown are specified by means of a MOF diagram. The code for these relations is written in the QVT-relational, it and the diagrams are shown in Figure 5.

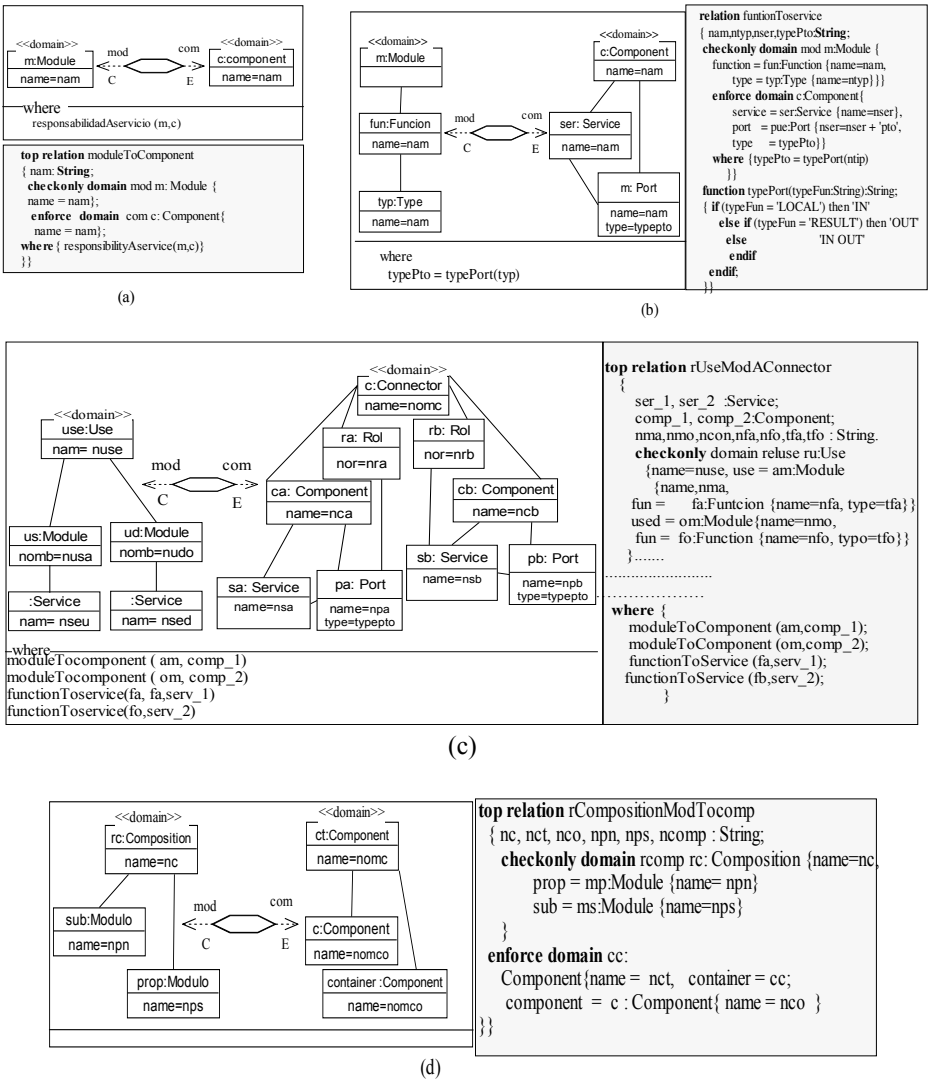


Fig. 5. Diagrams and codes of the relations: (a) moduleToComponent, (b) functionToservice, (c) rUseModToConnector, (d) rCompositionModTocomp

Relation *moduleToComponent*
This relation maps each module with a component. Figure 5(a) shown two types of relations: the *checkonly* type and the *enforce* type. The object of the *Module* domain is of the *checkonly* type. In contrast, the object of the *Component* domain is of the *enforce* type. This creates an object of the *Component* class that is related to the *Module* class. The *where* clause indicates a call to the *functionToService* relation, which relates an object of the *Module* class with an object of the *Component* class. The diagram and the corresponding code are shown in Figure 5(a).

Relation *functionToService*
This relation implies that a function of the module meta-model will generate a service of the Component class. When this occurs, the type of the Function will generate a *port*. In the *where* clause, the name of the *port* is obtained by calling the function *typePort*.

When the relation is executed, the classes that are in the source meta-model can only be verified and the classes that are in the target meta-model will be created (only the *ServiceToPort* class). The diagram and the corresponding code are shown in Figure 5(b).

Relation *rUseModToConnector*
The *uses* relation is transformed from the modular meta-class to a link between a connector and two components in the C&C meta-class. This is shown in Figure 5(c). The creation of objects for this relation is from 1 to *n* because a relation of two components is generated through a connector. Figure 5(c) shows part of the code to illustrate how the relations are invoked in the *where* clause to create the relations among module, component, and *functionToService*.

Relation rCompositionModToComp
The set of modules will also generate a set of components. However, in this case when a component is created (container) a subordinate is created (c) inside it. This is shown in Figure 5(d).

5 Applying Transformation at the Model Level

To illustrate how the transformations at the model level are applied in the transformation process (see Figure 3), the case study presented in [12] is used. This study consists of designing software architecture for a purchase and reservation ticket system for any kind of passenger transport (bus, air plane, train, etc) where the passengers can reserve and buy a ticket, or just search the Internet. After introducing the query, the system will generate several tentative itineraries as a first step. Next, the passenger can choose an itinerary and make the reservation or the purchase.

As a step previous to the transformation, the modular architectural view model is designed from the specification given above using the method proposed in [12]. The OMONDO tool developed for Eclipse [13] is used to design this model (and its corresponding meta-model) as shown in Figure 6. The modular view model is depicted in the window named *TicketPRModView* (Fig. 6, right). This model is made up of five classes (modules): *QuerForm, GeneIti, Purchase, Reservation,* and *RegisterAuthen*. These are all linked with each other through *use* relations. Next, the transformation

task for producing the C&C view model (target) is performed. To do this, the previously designed model is used as the source model. The relations established in section 5 are the rules that are applied to generate the new model (using the meta-model as templates). In this case, the *moduleToComponent* and *rUsoModToConnector* relations were applied. The transformation is carried out with the MOMENT [14] tool, which is a plug-in developed on ECLIPSE.

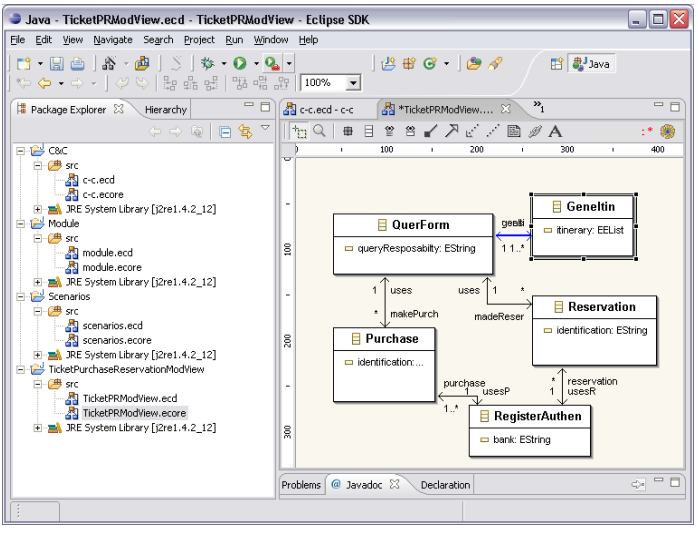


Fig. 6. The modular view model design of the reservation–purchase ticket system

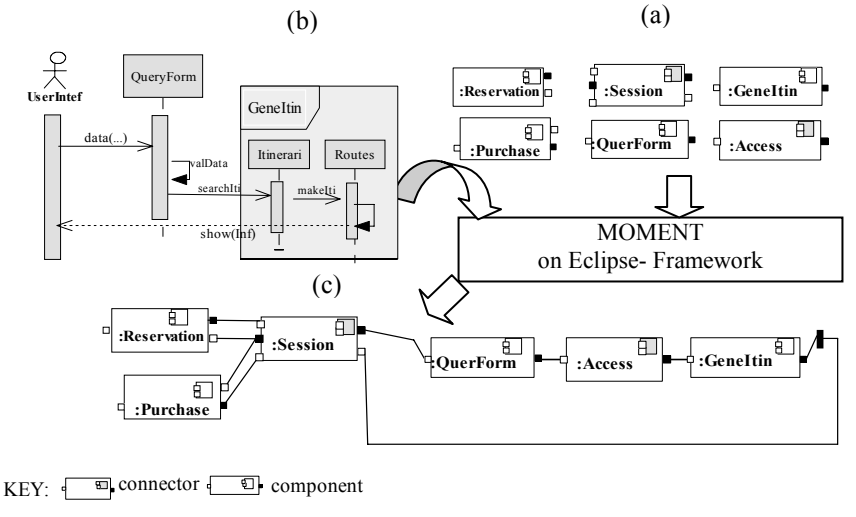


Fig. 7. The second transformation of models: (a) The C&C view model generated in the first transformation; (b) the scenario model; (c) The definitive C&C view model

The transformation generates the C&C view model shown in Figure 7(a). As this figure shows, a component was produced for each model, and each *use* relation generated a connector (*session and access*). However, these components and connectors are not linked with each other because their communication has not yet been modelled. A second transformation must be carried out to do this. The scenarios model is required for this second transformation. It is designed taking into account the requirement specification (not shown here). The designed model is shown in Figure 7(b) (partial).

The scenarios model and the C&C architectural view model are used as source models in this second transformation. As before, the MOMENT tool is used for this task; however, in this case, two source models are combined as shown in Figure 7. The final C&C architectural view model is then generated. This is shown in Figure 7 (c). This model and the modular architectural view model make up the software architecture of the purchase and reservation ticket system.

6 Conclusions

By modeling software architecture using MDA, we have proposed a novel way to design an architectural view. This approach can be extended to other types of views in software engineering.

The use of the transformation process among models proposed here, allows us to: a) establish meta-models for two views (modular and C&C), and identify and establish correspondence relations between them; b) generate a new view using another view as reference; c) refine the new view through a second transformation combining the new view and the scenarios model to obtain the C&C view model; d) adapt the MOMENT development tool to make this these transformations; e) apply a consistent transformation process attached to the standard specification of the OMG (MDA) by means of the QVT relation language and the MOF graphic notation to represent the relations and the transformation rules; f) adapt and use tools consistent with the language and notation used; g) apply a similar treatment to assure conformance between the C&C architectural view and the code (not included here); h) prove consistency (using rules *checkonly*) and contribute to improving the documentation.

This study has also shown that the application of MDA improves the design process of the software architectural views by doing it in a systematic and automatic way.

References

1. Bass, L., Clements, P., Kazman, R.: Software architecture in practice, 2[a] ed. Addison-Wesley, Reading (2003)
2. Shaw, M., Clements, P.: The Golden Age of Software Architecture. IEEE Software 23(1), 31–39 (2006)
3. Muskens, J., Bril, R.J., Chaudron, M.R.V.: Generalizing Consistency Checking between Software Views. In: Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05), pp. 169–180 (2005)
4. Fradet, P., Le Métayer, D., Périn, M.: Consistency checking for multiple view software architectures. In: Proceedings of 7th ACM SIGSOFT international symposium on Foundations of software engineering, pp. 410–428 (1999)

5. Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Nord, R., Stafford, J.: Documenting Software Architecture, Views and Beyond. Addison-Wesley, Reading (2002)
6. IEEE Product No. SH94869-TBR, IEEE Recommended Practice for Architectural Description of Software-Intensive Systems (2000)
7. Philippe, K.: The 4+1 View Model of Architecture. Paper published in IEEE Software 12(6), 42–50 (1995)
8. OMG, Object Management Group: Model Driven Architecture (MDA) (2004) http://www.omg.org/cgi-bin/doc?formal/03-06-01
9. Kleppe, A., Warmer, J., Bast, W.: MDA Explained: The Model Driven Architecture™: Practice and Promise. Addison Wesley Professional, Reading (2003)
10. OMG: Revised Submission for MOF 2.0. Query/View/Transformations RFP (ad/2002-04-10); OMG Document ad/2005-07-1 (2005)
11. Unified Modeling Language 2.0. Superstructure, http://www.omg.org/docs/formal/05-07-04.pdf
12. Limon, C.R., Ramos, S.I., Torres, J.J.: Designing Aspectual Architecture Views in Aspect Oriented Software Development. In: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganà, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3983, pp. 726–735. Springer, Heidelberg (2006)
13. Eclipse: http://www.eclipse.org/
14. The MOMENT Project, http://moment.dsic.upv.es/

Update Propagation Technique for Data Grid

Mohammed Radi[1], Ali Mamat[1], M.Mat Deris[2], Hamidah Ibrahim[1], and Subramaniam Shamala[1]

[1] Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
Moh_radi@alaqsa.edu.ps, Ali@fsktm.upm.edu.my
[2] Faculty of Information Technology and Multimedia University of Tun Hussein Onn, P.O. Box 101, 86400 Parit Raja, Batu Pahat, Johor, Malaysia
mmustafa@kuittho.wdu.my

Abstract. Data replication is a well known technique used to reduce accesses latency, improve availability, and performance in a distributed computing environment. An asynchronous replication is a commonly agreed solution for the consistency of replicas problem. Update propagation using a classical propagation schema called the radial method suffers from high overhead of the master replica while line method suffers from high delay time. This paper presents a new asynchronous replication protocol called Update Propagation Grid (UPG) which especially for a wide area distributed Data Grid. Updates reach other replicas using a propagation technique based on nodes organized into a logical structure network that enables the technique to scale well for thousands of replicas. Restructuring operation is provided to build and reconfigure the UPG dynamically. An analytical model is developed; communication cost, average load balance, and average delay time have been analyzed. The technique achieves load balancing and minimizes the delay for file replication in Data Grid.

1 Introduction

Many large-scale scientific applications such as high energy physics, data mining, molecular modeling, earth sciences, and large scale simulation produce large amount of datasets [1, 2] (in order of several hundred gigabytes to terabytes). The resulting output data of such applications need to be stored for further analysis and shared with collaborating researchers within scientific community who are spread around the world. These requirements are related to Data Grid application. As in most distributed environments [3] Data Grid has been used replication to reduce access latency, improve data locality, increase robustness, scalability and performance for distributed applications. Several data replication techniques [4, 5, 6, 7, 8 ,9] have been developed to support-high-performance data access, improving data availability, and load balancing to remotely produced scientific data. Most of those techniques do not provide the replica consistency in case of update. Grid data management middleware usually assumes that (1) whole files are the replication unit, and (2) replicated files are read only. However there are requirements for mechanisms that maintain consistency for a

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, pp. 115–127, 2007.
© Springer-Verlag Berlin Heidelberg 2007

modifiable data. In general, the data consistency problem deals with keeping two or more data items consistent, but in data grid consistency is keeping replicas up to date [10]. The main issue in Data Grid system is to maintain scalability with highly dynamic and large number of replicas distributed over wide area networks while maintaining the same view of all replicas. In distributed database as well as in distributed file systems, content distribution networks web applications; several solutions of replica synchronization already exist by using optimistic consistency protocols [11]. Those solutions are only partially applicable to the problem in Data Grid infrastructure. An asynchronous replication is more suitable for distributed environment. A few studies have been done to maintain replica consistency in Data Grid [12, 13, 14, 15, 16]. Most of the recently techniques do not take into account the main characteristics of data grid, mainly the high number of replica sites distributed in different domain, and the highly dynamic environment. Besides those techniques use classical propagation schemes, which we call a radial scheme, in which the master sends the data to all replica sites. Sending an update from the master replica to each other replica may have high cost, the same message may travel over the same communication link repeatedly and tie up the resource at the master site for a very long period of time, moreover granting reliability of delivery become more difficult, network portioning may happen and individual host may fail. The second approach that can be used to propagate the update is called line approach, in which the master replica sends the update information to only one site and this site sends the update information to only one site and so on. It may take a long period of time for an update to propagate to all other host. Recently the update issues have been discussed in P2P systems [17], which seem to be similar to the issue in Data Grid. This paper focus on modeling an update propagation technique based on asynchronous replication to increase the system performance, such as load balancing, and delay reduction.

The rest of this paper is organized as follows: in section 2 we discuss the replica consistency framework. In section 3 the propagation technique is proposed. Section 4 the analytical model is derived. In section 5 the performance analyses and comparison with radial and line methods are given. The conclusion presented in section 6.

2 Replica Consistency in Data Grid

2.1 System Model and Assumption

We consider a data sharing model in a federated organization in which replica consistency catalog has information about the sites that replicate the same data. Furthermore we assumed a large number of grid sites and each grid site infrequently join and leave the system, and also an active grid site is able to communicate with any other online site. We assume that the update information is needed to immediately notify to all active sites storing the replica when the update occurs. We assume that replica network to be widely distributed with no internal structuring. As our focus is on virtual network then the physical connectivity and system topology were not considered.

The basic idea of a single master replication is that for each data item or file there exist one master copy and all other replicas are secondary replicas. The user updates only pass to the master replica, which does the updates and then propagates the changes in form of update information to all secondary replicas. The secondary sites receive the update information and apply the modification to its home replica. To provide the required functionalities of the single master replication, Replica Consistency Service (RCS) architecture is proposed. Local Replica Consistency Service (LRCS) and Replica Consistency Catalogue (RCC) are the main components of the architecture. Two types of LRCS are proposed; the first one is master replica which is the main entry of the update requests from users, updates its local replica, generates the update information unit and begins the update propagation process. The second one in the secondary replica and it is responsible for update its local replica, relays the update propagation process if necessary, as well as the registrations and withdraws of its local replicas. RCC use to store the metadata use by the RCS, and to manage the UPG. The interactions of the above components are shown in Figure 1.

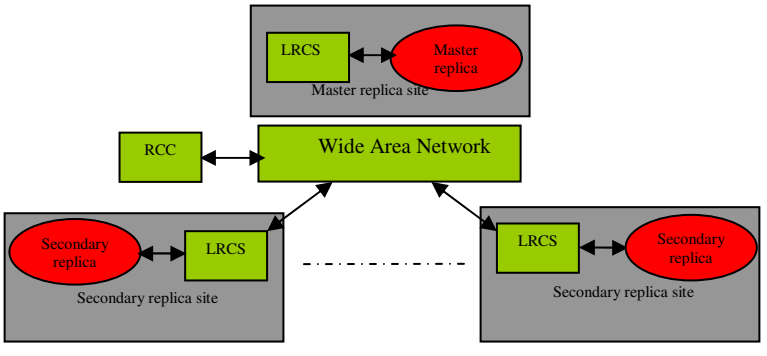


Fig. 1. The replica consistency service framework

2.2 The Construction and Maintenance of Update Propagation Grid (UPG)

In this section how to construct and maintain the UPG will be discus. In Data Grid First the master copy of the file is initiated, after that the file can be searched, fetched and replicated by another sites. RCC maintain information about each replication process. From the replication information replica tree may naturally constructed.

Figure 2 shows a replica tree composes of 10 sites and the root of the tree is the master replica stored in site number 1. If all sites are always on-line, and none of the replica sites allow deleting the replica, an update from the master site can be successfully propagated to any other sites through the replica tree. For example when site 1 initiated the update, it sends the update information to site 2, 3, and 4 and then site 5, 6 receive the update from site 2. Each replica site updates its home replica and then relays the update to all its children. Then the update is successfully propagated through all the sites.

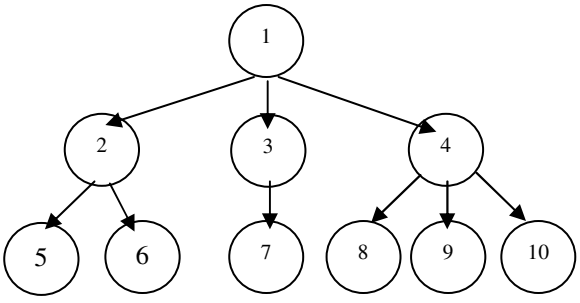
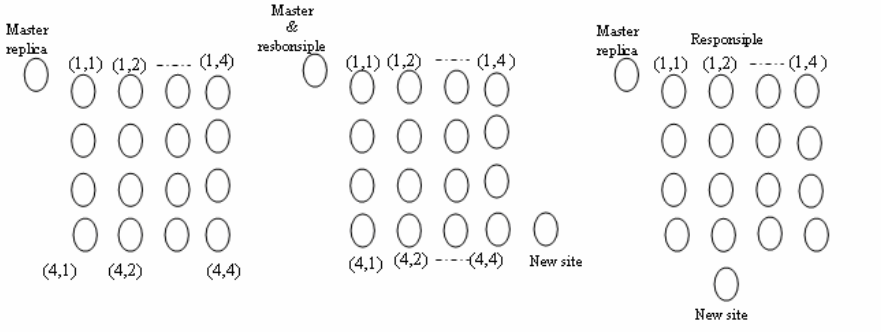


Fig. 2. Replica site constructs a replica tree

Highly dynamic environment such as Data Grid, site may connect and disconnect at any time, and the site may delete, and add replica at any time. Such a replica tree will be ineffective in term of update delivery. In order to increase the probability of successfully propagating the update, each time a site deletes or adds a replica, the replica tree for its copies must be modified and restructured. Moreover due to the properties of general tree, some site may maintain the information of a large number of sites, while some other maintain for a very few number of site, this will unbalance the overhead associated with the file maintained by each site. In order to achieve better performance and load balance, and delay reduction for the update propagation process an Update Propagation Grid (UPG) is constructed from a replica tree as explained in section 2.3.

2.3 Participation of a New Site to UPG

In UPG all replica sites are logically organized in the form of two-dimensional grid structure. For example as shown in figure 3(A) is UPG consists of 16 replica sites. The sites are organized in the form of 4X4 grids. Each site has one master replica and the other sites are secondary replicas. We calls each site as UPG (I, J), where I refers to the rows, and J refer to the columns. When a site makes a copy of a data replica, it participates in the UPG of the data replica, and we call this site as new site. We also call the site that determines the location of the new site in the UPG as a responsible site. The responsible site is the site from which the new site creates a replica from. If the responsible site is the master copy site then the new site will create a new column in the UPG. If the responsible site is a secondary copy then the new site is added in the column in which the responsible site is hold. Figure 3 shows two basic examples of the participation of new site. The initial state of a given data set UPG is shown in figure 3(A). Suppose now new site copies a replica, then its LRCS asks the RCC to participate to the replica UPG by sending the responsible site information. In the first case the new site copies its data item from the master replica, then the new site position will be in the new column as shown in 3(B). And in the second case the new site copy its data idem from a site G (1, 2) then the new site position will be in the column 2 as shown figure 3(C).



(A): the initial UPG

(B): participation when the responsible is the master

(C): participation when the responsible is not the master

Fig. 3. Participation new site in the replica UPG

2.4 Maintain the Consistency of UPG

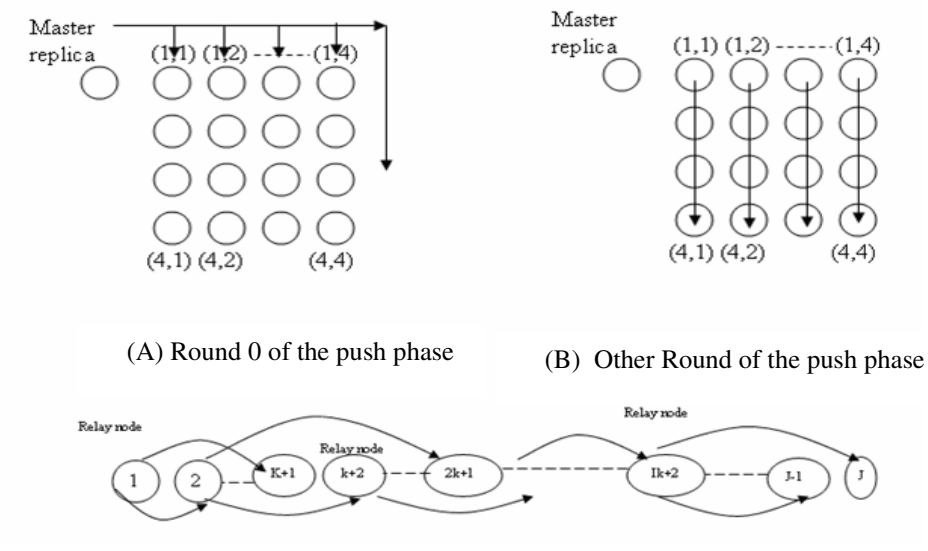
Generally grid sites have limited data storage and thus it often occurs that some replicas must be deleted by the grid system for preparing sufficient space for new replica, or the replica doesn't use anymore locally, or no request has been made to access remotely from another sites after a certain amount of time, the system decides to delete the replica. Moreover replica may be removed from the site when site user chooses to remove it. When the replica is deleted the local LRC has to inform the RCC by the deletion process information and the RCC has to maintain the UPG, for this replica by removing the site information from the replica UPG.

Grid is highly dynamic environment so the sites are frequently disconnected from the Data Grid system. In order to increase the performance of the update propagation process and reduce unnecessary communication cost we need to have consistent information about each grid site. Strong consistency is not required in environment such as data grid [4], so RCC must receive periodic status message from LRCS describing their states. Each LRCS must periodically refresh its information to show its replica is still available. On that time if the LRCS does not send its refresh information then RCC will recognize that the LRCS is not available and now have to change its replica status to be 0, mean not available now. And no need to send unneeded message.

3 Update Propagation Techniques

3.1 Push Phase

When master site user initiates an update to master copy, the LRCS inquires the RCC to send the UPG information. The LRCS starts the update propagation process after receive the UPG information from the RCC. The update propagation process is shown in figure 4 as shown in 4(A) the round 0 of the update propagation in which the master replica site sends the update information to the first site in each column. The first site in each column propagates the update information in parallel with other first sites, as shown in figure 4(B). Figure 4(c) gives details of the other round in each column.



(A) Round 0 of the push phase

(B) Other Round of the push phase

(C) Detail of the other rounds of the push phase for the column i

Fig. 4. The push phase of the update propagation process

This process is done recursively until all sites in the column have the update information and update propagation process will be done in parallel in each column.

3.2 Pull Phase

During the offline period of a Data Grid site, it may miss some update of the file. Hence when the offline node reconnected, it needs first to modify its UPG information in the RCC, then the RCC sends to the reconnected site its responsible site . The responsible site compares its home replica timestamp with the replica timestamp send by the reconnected site, if the replica timestamp is less than the home timestamp, the responsible site send the update information to the reconnected replica.

4 Performance Analysis

The goal of UPG model is to achieve replica consistency, through update propagation process. UPG model is able to achieve load balance, delay reduction and less communication cost. The average delay is the average total hop count from the master copy to each site that receives the update information. The average load balance is the average number of site to which each grid site propagates the update information.

An analytical model is developed in this section; the communication cost, average load balance, and average delay time will be analyzed. In the analysis we start from a completely consistent state and analysis a single update request. Average delay time,

Table 1. Parameters and measurement matrices

N	Number of replica sites
I	Number of columns
J	Number of sites in each column (number of rows)
K	Number of relay node
U	The size of the update message
B	Size of data required to describe one replica
ML(r)	The size of message in round r
R	Number of rounds
Msg(r)	The number of needed in round r
D	The update propagation delay (in hop)
AVGD(N)	Average delay time of N replicas
AVGL(N)	Average load balance of N replicas

the average load balances and number of messages required to reach a consistent state have been evaluated. For the purpose of easy analysis we assume that the replica sites in UPG will form a two dimensional Grid. Table 1 shows some parameters and measurement matrices use in the analysis.

4.1 Average Delay Time

The primary performance criterion is the delay throughput characteristic of the system. In our UPG we are concerning the delay time in a virtual network, so we will consider the time of communication between two grid sites equal to one time unit as a constant model T = 1. And we can defined the delay time of an update information to a given grid site as the number of hop count from the master replica to that grid site. The average delay time for the radial method is AVGD (N) =1, and for the line method is AVGD (N) = N/2. To analyze the UPG method for N nodes we will assume that the nodes formulate a rectangular grid UPG (I, J), where N =I× J.

Theorem 1. *I*n an N replica sites forming an UPG (I, J), with K relay node, the average delay time of the UPG algorithm is:

$$\mathrm{AVGD(N)} = \frac{I\left[1+\frac{K}{2}\left(\frac{(j-1)}{K}-1\right)\left(\frac{K-1}{K}+2\right)\right]}{N} \tag{1}$$

Proof. In the first round the master site send the update information to I nodes. The delay time for the I nodes is 1. At the second round each one of the (I) relay node send the update information to K nodes. The delay time of the K×I nodes is 2, in the third round same to the second round, we have K×I nodes with delay 3, at the M round we have K×I nodes with delay time M. the number of round is R =(J-1)/K round.

The total delay time (D) is

$$D = I + 2IK+3IK+\ldots\ldots\ldots \left(\frac{J-1}{K}\right) I \times K \tag{2}$$

$$=I[1+(2K+3K+\ldots\ldots\ldots \frac{J-1}{K} K)] \tag{3}$$

$$= I\,[1+K\sum_{h=2}^{\frac{J-1}{K}} h\,] \tag{4}$$

$$= I\left[1+\frac{K}{2}\left(\frac{(J-1)}{K}-1\right)\left(\frac{K-1}{K}+2\right)\right] \tag{5}$$

$$\text{AVGD (N)} = \frac{I\left[1+\frac{K}{2}\left(\frac{(J-1)}{K}-1\right)\left(\frac{K-1}{K}+2\right)\right]}{N} \tag{6}$$

4.2 Average Load Balance

The performance of any system designed to exploit a large number of computers depends upon the balance distribution of workload across them. For the update propagation process this will be achieved by distribute the overhead of the update propagation process over many nodes. We defined the average load balance as the average number of sites to which each site propagates the update information. The average load balance for the radial algorithm is AVGL (N) = N, and for line algorithm is AVGL (N) = 1.

The average load balance of the UPG is at most K because each site propagates the update information to not more than K sites. The number of nodes shares in the update propagation process is

$$I\times\left(\frac{J-1}{K}\right) \tag{7}$$

The average load balance of the UPG will be

$$\text{AVGL (N)} = K \tag{8}$$

4.3 Communication Cost

For UPG (I, J) with N grid sites, we are going to define the communication cost by determines the number of messages to implement the update propagation process until

reach a consistent state. We prove that the UPG will not exceed messages more than the line and the radial approaches.

Theorem 2. The total number of messages to push an update to N replica is N messages.

Proof: In round (0), the master replica site sends the update information message to I replicas. Thus we obtain a total number of messages.

$$\text{Msg (0)} = \text{I}. \tag{9}$$

In round (1) each relay site sends one message to k replica holders. We have

$$\text{Msg (1)} = \text{I} * \text{K messages}. \tag{10}$$

The number of rounds is $\left(\frac{J-1}{K}\right)$

From round 1 to round $\left(\frac{J-1}{K}\right)$, in each round we send I* K messages. The total number of messages from round 1 to round $\left(\frac{J-1}{K}\right)$ is

$$\text{Msg (1...}\left(\frac{J-1}{K}\right)\text{)} = \sum_{g=1}^{\frac{J-1}{K}} kI = \left(\frac{J-1}{k}\right) \times \text{I} \times \text{K} \tag{11}$$

$$= (J-1)I = J \times I - I \tag{12}$$

The total number of messages from round 0 to round $\left(\frac{j-1}{k}\right)$ can be calculated by summation of equations 9 and 12.

$$\text{Msg (R)} = JI - I + I \tag{13}$$

$$\text{Msg (R)} = JI = N \tag{14}$$

From the pattern of the update propagation process described in section 3, we can see that every nodes can receive the update information only once, and the maximum number of messages to push an update is N.

5 Numerical Result and Comparisons

Based on the analytical model has been developed in the section 4 we first evaluate the UPG method and compare it with the line and radial methods for load balance and

average delay time. Then investigates the effects of varying the K. the number of messages have not been compared because in the three techniques the number of messages is equal to the number of replica sites.

5.1 Evaluation When K=2

First we examine the average load balance and the average delay of each of the three methods (line, radial, UPG where k=2). For the UPG case we set number of column to be 5. We compare the three methods with varying the number of N, where N= 100×h (h=1, 2…10).

The horizontal axis in figures 5, 6 indicates the number of sites in the networks. The vertical axis indicates average load balance in figure5, and average delay time in figure 6. In figure 5, the average load balance for the line propagation technique is always 1 because each site propagates the update information to another one replica site. The average load of the radial propagation is always equal to the number of replica sites. The results are considering the characteristics of the two methods. On other hand the average load balance of UPG method is at most 2, because in UPG method each site propagates the update information to not more than 2 sites.

As shown in figure 6, the average delay time of the radial propagation method is always 1 because the master replica directly propagates the update information to all sites holding the replica. The average delay of the line propagation method linearly gets higher. Also these results are obvious considering the characteristics of the two methods. On the other hand the UPG method keeps the average delay linearly increase but in a very low fraction than the line method.

From the results, it has been shown that the UPG method reduces the average load compared with the radial propagation method and reduces the average delay compared to the line propagation method. Thus we can confirm that the UPG achieves both load balance and delay reduction for update propagation in Data Grid network.

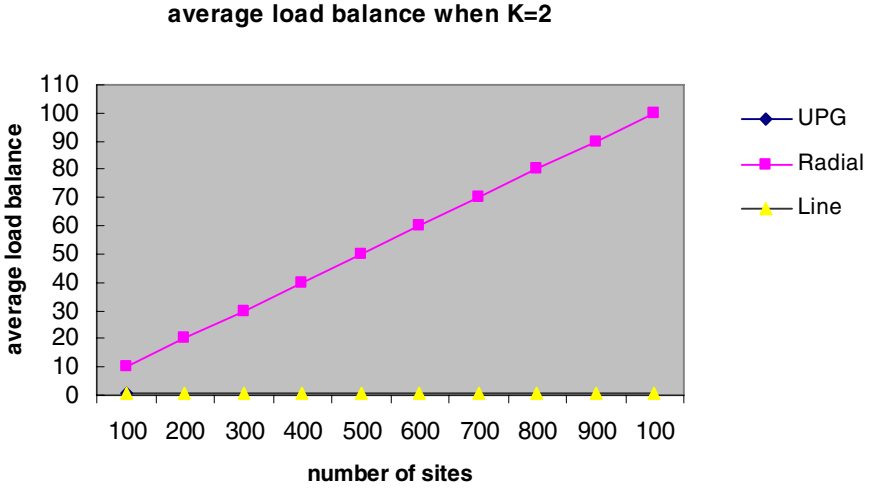


Fig. 5. Average load balance where k=2

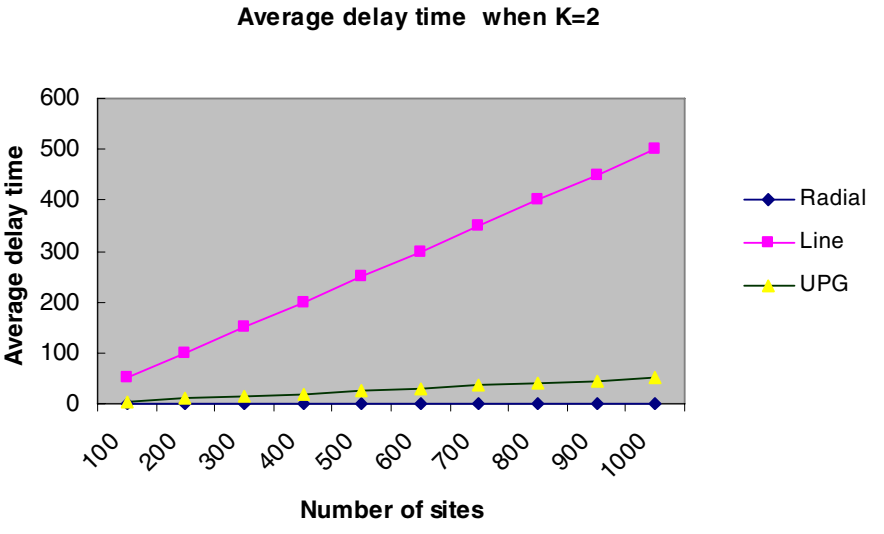


Fig. 6. Average delay time where k=2

5.2 An Effect of K, Evaluation Where K=2, 3, 4, …, 10

The numerical result is presented to characterize the optimal value of K under different number of sites. We studied the impact of K in the average load balance and average delay time. In this case we set the number of replica sites N to be 100*h (h=1, 2,…, 10). And the number of column is 5 and varying the K from 2, 3,…, 10.

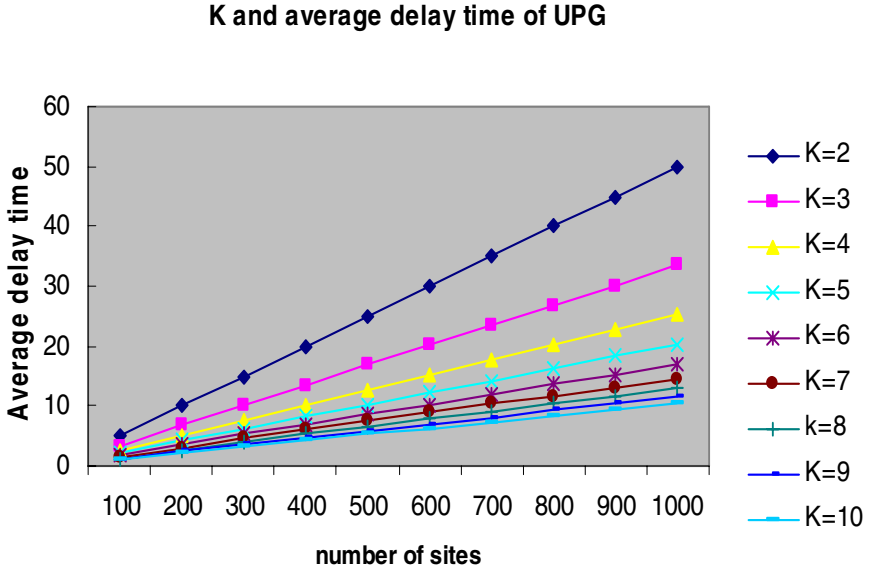


Fig. 7. K and average delay time

Figure 8 and 9show the average load and average delay of the UPG method, when varying the value of K, from 2 to 10. Figure 8 shows that the average load balance increases as the number of K get higher. Figure 7 shows that show that as K get larger the average delay time get higher. We can balance the average load and the average delay by changing K to get better result, depending on the sites distribution and number of site.

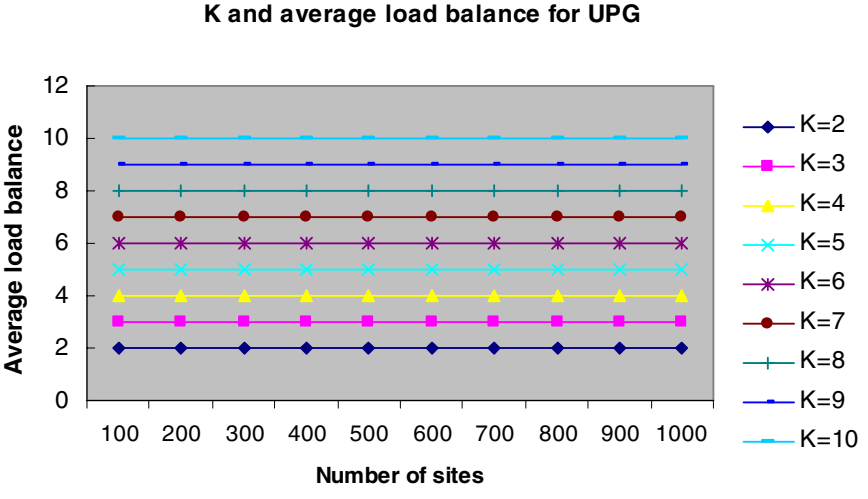


Fig. 8. K and average load balance

6 Conclusion

In this paper, a new technique called Update Propagation Gird (UPG) has been proposed to improve the load balance and delay time of the update propagation process in an asynchronous environment. We assume a Data Grid environment where the update information is immediately notified to all sites when the update occurs. Update Propagation Grid (UPG) is logical structure network enables the scheme to scale well for thousands of replica and minimize the communication cost, while ensuring reliable delivery. In our proposed technique updates reach other replicas using a propagation scheme based on UPG. Restructuring operation is provided to build and reconfigure the UPG dynamically. Moreover we verified the effectives of the proposed method by the analytical analysis for the number of messages, the average load balance and the average delay. The numerical results show that our method gives better average delay than the line method, and gives better average load balance than the radial method.

References

1. Chervenak, I., Foster, C., Kesselman, C., Salisbury, C., Tuecke, S.: The data grid: Towards architecture for the distributed management and analysis of large scientific datasets. Journal of Network and Computer Applications 23, 187–200 (2001)
2. Allcock, J., Bester, J., Bresnahan, A.L., Chervenak, I., Foster, C., Kesselman, S., Meder, V., Nefedova, D., Quesnal, Tuecke, S.: Data management and transfer in high performance computational grid environments. Parallel Computing Journal 28(3), 749–771 (2002)

3. Gray, J., et al.: The Dangers of Replication and a Solution. In: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, pp.173–182 (1996)
4. Chervenak, A., et al.: Giggle: A Framework for Constructing Scalable Replica Location Services. In: Proc. Of SC 2002 (2002)
5. Samar, H S.: Grid Data Management Pilot (GDMP): A Tool for Wide Area Replication. Applied Informatics (AI) (2001)
6. Cameron, J., Casey, L., Guy, P., Kunszt, S., Lemaitre, G., McCance, H., Stockinger, K., Stockinger, G., Andronico, W., Bell, I., Ben-Akiva, D., Bosio, R., Chytracek, A., Domenici, F., Donno, W., Hoschek, E., Laure, L., Lucio, P., Millar, L., Salconi, B., Segal, M.S.: Replica Management Services in the European Data Grid ProjectUK. e-Science All Hands Conference, Nottingham (September 2004)
7. Bosio, Bell, W.H., Cameron, D., McCance, G., Millar, A.P., et al.: EU Data Grid Data Management Services ,UK e-Science All Hands Conference, Nottingham (September 2003)
8. Lamehamedi, H., Szymanski, B.: Decentralized Data Management Framework for Data Grids. Future eneration Computer Systems 23(1), 109–115 (2007)
9. Guy, L., Kunszt, P., Laure, E., Stockinger, H., Stockinger, K.: Replica Management in Data Grids. Technical Report, GGF5 Working Draft, Edinburgh Scotland (July 2002)
10. Dullmann, W.H., Jaen-Martinez, J., Segal, B., Stockinger, H., Stockinger, K., Samar, A.: Models for Replica Synchronisation and Consistency in a Data Grid. hpdc. In: 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10 '01), San Francisco, CA, USA, August 7-9, 2001, p. 67. IEEE Computer Society Press, Los Alamitos (2001)
11. Venugopal, S., Buyya, R., Ramamohanarao, K.: A Taxonomy of Data Grids for Distributed Data Sharing. Management and Processing. In: ACM Computing Surveys, vol. 38(1), pp. 1–53. ACM Press, New York, USA (2006)
12. Domenici, A., Donno, F., Pucciani, G., Stockinger, H., Stockinger, K.: Replica Consistency in a Data Grid, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 534(1-2), 24–28 (2004)
13. No, J., Park, C., Park, S.: Data Replication Techniques for Data-Intensive Applications. In: International Conference on Computational Science. vol. 4, pp. 1063–1070 (2006)
14. Chang, R., Chang, J.: Adaptable Replica Consistency Service for Data Grids. In: Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06) (2006)
15. Sun, Y., Xu, Z.: Grid Replication Coherence Protocol. ipdps. In: 18th International Parallel and Distributed Processing Symposium (IPDPS'04) - Workshop. vol. 13, p. 232b (2004)
16. Domenici, A., Donno, F., Pucciani, G., Stockinger, H.: Relaxed Data Consistency with CONStanza. In: CCGRID 2006, pp. 425–429
17. Wang, Z., et al.: An efficient update propagation algorithm for P2P systems, Comput. Commun.(2006), doi:10.1016/j.comcom.2006.11.005.

A Spatiotemporal Database Prototype for Managing Volumetric Surface Movement Data in Virtual GIS

Mohd Shafry Mohd Rahim[1], Abdul Rashid Mohamed Shariff[2], Shattri Mansor[2], Ahmad Rodzi Mahmud[2], and Daut Daman[1]

[1] Faculty of Computer Science and Information System, University Technology Malaysia, 81310 Skudai, Johor, Malaysia
[2] Institute of Advance Technology, University Putra Malaysia, 43400 Serdang, Selangor, Malaysia
shafry@utm.my, rashid@eng.upm.edu.my
shattri@eng.upm.edu.my, arm@eng.upm.edu.my, daut@utm.my

Abstract. Virtual GIS have the capability to present geographic features in three-dimension. This will contribute to increasing user understanding of phenomena that occurred in certain areas. Virtual GIS need to include temporal element in the system for managing dynamic phenomena to simulate real process. Spatiotemporal database system can be used to manage temporal data or spatial data that involve changes. In this paper, we present the database system of a spatiotemporal database prototype for managing volumetric surface movement; Volumetric Surface Movement Spatiotemporal Data Model that has the ability to store changes of the surface data of a three dimensional object. Formalization of the surface movement phenomena on the volumetric object were defined. This model has been implemented using relational database model which store only the changing points in order to ensure an efficient data management. We have developed a visualization tool with the database system to visualize movement on the surface. The prototype system of this model has been tested and found to be efficient.

Keywords: Spatiotemporal Data Model, Database, Surface Movement, Temporal GIS, Virtual GIS.

1 Introduction

Virtual Geographical Information Systems (VGIS) is generally defined as the use of computer graphic technology to improve the presentation of geographical information. Therefore, Geographical Information Systems (GIS) will be able to become more realistic with real process and presentation precisely as in the real world. In many cases, simply introducing an additional orthogonal axis (Z) is convenient but undeniably insufficient. This is because important spatial and temporal characteristics and relationships may be imperceptible in this approach. Although visualization techniques for three or more dimensions have become popular in recent years, data models and formal languages have not yet

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 128–139, 2007.
© Springer-Verlag Berlin Heidelberg 2007

fully developed to support advanced spatial and temporal analysis in multiple dimensions [1].

In Virtual Geographical Information Systems (VGIS), data model is the abstraction of the real world phenomena according to a formalized, conceptual scheme, which is usually, implemented using the geographical primitives of points, lines, and polygons or discredited continuous fields [2]. Data model should define data types, relationships, operations, and rules in order to maintain database integrity [2]. In VGIS, data model is also used to enhance the focus on 3D data. Thus, spatiotemporal data model in VGIS is an abstraction of managing 3D spatial with temporal elements. Spatiotemporal data model is very important in creating a good database system for VGIS which deals with space and time as main factors in the system [3,15].

A variety of spatiotemporal data model were developed previously and for the purpose of this research, we have collected and analyzed 9 data models namely, GEN-STGIS [5], Cell Tuple-based Spatiotemporal Data Model [6], Cube Data Model [7], Activity-based Data Model [8], Object-based Data Model, Data Model for Zoning [9], Object Oriented Spatial Temporal Data Model [10], Multigranular Spatiotemporal Data Model [11] and Feature-Based Temporal Data Model [12].

We agree that in order to create a VGIS with a realistic process, an appealing spatiotemporal data model should focus on the volumetric data and geographical movement behavior. Based on our analysis, an issue which is related to the capability of spatiotemporal data model is the 3D visualization of volumetric spatiotemporal object. This capability is vital in order to increase user understanding in the geographic phenomena so to create simulations or future predictions. Thus, data model in VGIS must have the capability of user query to visualize information in the form of 3D with movements. This is indeed a very challenging issue. It has also been addressed by [13] that the current development of techniques and tools are simply unable to cope when expanded to handle additional dimensions. In this paper, we will discuss implementation of our proposed data model which is the Volumetric Surface Movement Spatiotemporal Data Model. This model was developed to manage movement of the volumetric object with less data redundancy, and also to retrieve data for visualization. Our discussion will focus on the system architecture in general which includes database development, data retrieval and visualization approach.

2 System Architecture

Figure 1 shows the architecture of our prototype system which consists of *Data Storage, Spatiotemporal Data Loader Processor, Spatiotemporal Data Retrieval Processor* and *Volumetric Surface Movement Simulator*.

Data Storage stores the volumetric surface movement object according to the database scheme. The database that were used in this system is MySQL which is open source. *Spatiotemporal Data Loader Processor* provide methods of loading data into the database. This processor will ensure no data redundancy in the database by storing only the points which were involved during the surface movement.

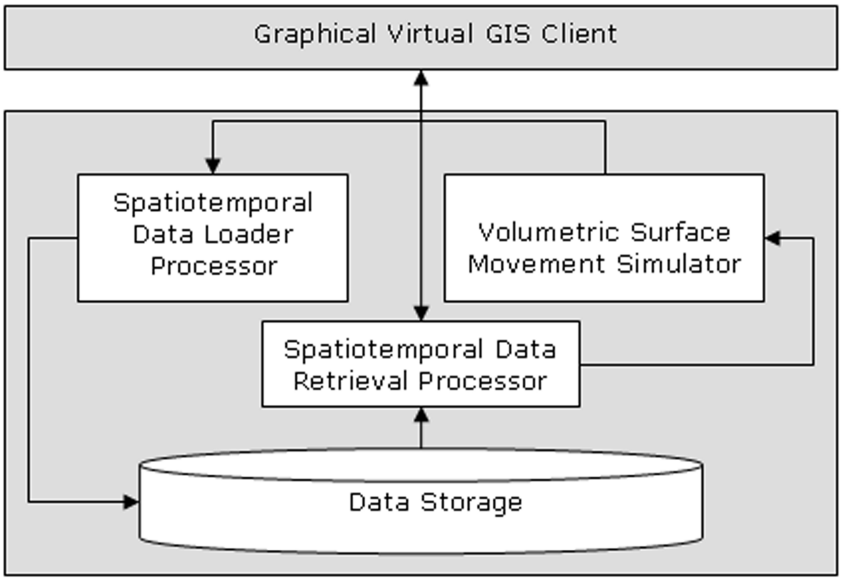


Fig. 1. Architecture of the system

Spatiotemporal Data Retrieval Processor will retrieve the data from the database based on what the user requires. In this case, the user need to input the area and duration of data. Data that were retrieved will be transferred into our own file format. *Volumetric Surface Movement Simulator* will read the retrieved data set and simulates the volumetric surface movement object. *Graphical Virtual GIS Client* accepts users queries, and sends it to the *Spatiotemporal Data Retrieval Processor*, and illustrate the query result in graphical presentation.

3 Volumetric Surface Movements Spatiotemporal Data Model

Based on the volumetric surface movement formalization by [4] we can see that conceptually the data that were used to define the surface of a volumetric object are actually the points with the value of x, y, and z. This refers to the location and the coordinate of the points. Whenever there is a change on the surface, these points also change. Therefore time is the utmost important component in efficiently managing the points that forms the surface.

The model is based on the object representation (1) and (2) below which relates to the data that forms the surface and are stored and extracted using the equation of $f(mv)t_1, t_2, \ldots, t_m$ where it is a combination of the surface set. The combination from the surface set is classified as a volumetric surface movement object. This will be the base in building a data model to manage data that involves surface movements and the foundation in managing volumetric surface movement object in the real world.

$$f(mv)t_1,\ t_2,\ \ldots\ ,\ t_m \rightarrow f(v,\ t_1) \cup f(v,\ t_2) \cup \ldots \cup f(v,\ t_m) \quad (1)$$

$$f(mv)t_1,\ t_2,\ \ldots\ ,\ t_m \rightarrow [\ \{\ <x_1,y_1,z_1,\ t_1>,\ <x_2,y_2,z_2,\ t_1>,\ \ldots<x_n,y_n,z_n,\ t_1>,\ \}$$
$$\cup\ \{\ <x_1,y_1,z_1,\ t_2>,\ <x_2,y_2,z_2,\ t_2>,\ \ldots<x_n,y_n,z_n,\ t_2>,\}$$
$$\cup\ \ldots\ \cup\ \{\ <x_1,y_1,z_1,\ t_m>,\ <x_2,y_2,z_2,\ t_m>,\ \ldots<x_n,y_n,z_n,\ t_m>,\ \}\ ] \quad (2)$$

In real process, not all of the points in the volumetric surface moves or changes. This raises the question as to whether it is necessary to store all of the points, which will increase the storage usage in the implementation. Therefore, in order to avoid data redundancy, the data model must be able to identify which point that has changed. To identify, data model must have the capability to check every point among version of data and capture the changing point. To perform this task, the formalism that was defined earlier will be used. The conceptual identification is as follows:

Assume $Point_1$ *is* $<x_1,\ y_1,\ z_1>$ *at time* t_n
$Point_2$ *is* $Point_1$ *at time* t_{n+1}
If $Point_2$ *-* $Point_1 = 0$ *then* $Point_2 = Point_1$ *at time* t_{n+1}
If $Point_2$ *-* $Point_1 \neq 0$ *then* $Point_2 = Point_2$ *at time* t_{n+1}

Visually, our proposed data model can be translated in table form to give a clearer understanding. Table 1 below describes the proposed data model.

Table 1. Volumetric Surface Movement Data Model

v/t	**P1**	**P2**	**P3**
tn	(x1,y1,z1)	(x2,y2,z2)	(x3,y3,z3)
tn+1	(x1,y1,z1) || (x1,y1,z1)'	(x1,y1,z1) || (x1,y1,z1)'	(x1,y1,z1) || (x1,y1,z1)'
.	.	.	.
.	.	.	.
tn+m	(x1,y1,z1)" || (x1,y1,z1)"'	(x1,y1,z1)" || (x1,y1,z1)"'	(x1,y1,z1)" || (x1,y1,z1)"'

Each point will go through changes but if there is a change that occurs at time t, not all of the points that form the surface changes. Therefore to store data at time t, a comparison will be done to determine which points actually changes. Thus, the process identification as stated in algorithm above is used. The process is as follows:

$P1\ (x_1,y_1,z_1,t_n)\ move \rightarrow P1'\ (x_1,y_1,z_1,t_{n+1})\ move \rightarrow \ldots \rightarrow P1'\ (x_1,y_1,z_1,t_{n+m})$
If (x_1,y_1,z_1,t_{n+1}) *-* $(x_1,y_1,z_1,t_n) = 0$
$(x_1,y_1,z_1,t_{n+1}) = (x_1,y_1,z_1,t_n)$
Else if $(\ (x_1,y_1,z_1,t_{n+1})$ *-* $(x_1,y_1,z_1,t_n) > 0\ )\ \|\|$
$((x_1,y_1,z_1,t_{n+1})$ *-* $(x_1,y_1,z_1,t_n) < 0)$
$(x_1,y_1,z_1,t_{n+1}) = (x_1,y_1,z_1,t_{n+1})$

Therefore, the next point will be stored in the storage for every point after movement occurs

$$from\ t_n\ until\ t_{n+1} \rightarrow (x_1,y_1,z_1) \parallel (x_1,y_1,z_1)'$$

By executing this process, movement data can be stored more easily while avoiding data redundancy. Hence, changes on a volumetric surface were managed more efficiently.

4 Database System Development

In Volumetric Surface Movement Spatiotemporal Data Model or VSMST Data Model, surface on the volumetric object are represented on a 3D space together with the temporal element which represents the changes or movement. To specify their position in space, this data model uses Cartesian frame for reference. This frame is chosen based on the successful development of the database system in GEN-STGIS by [5]. More specifically, VSMST Data Model uses 3 dimensional Cartesian (x,y,z) coordinate system and it is based on the Euclidean geometry, which concerns the study of points, lines, polygons and other geometric figures that can be formed from it (i.e. polygons, polylines, and arcs) [5] in an Euclidean plane. The VSMST Data Model is based on the linear model of time, where time is represented as a discrete variable. This is based on the definition in the previous section which defines behavior of time that relates to the surface movement. In reality, events are recorded in a discrete manner, hence time can be represented in the data model. The relational approach has been used to design and implement the VSMST Data Model to ensure that it is flexible enough so that this model can be implemented in various database systems in the market. In this section, we will discuss the development of the database to manage and visualize surface movement data in a volumetric surface.

4.1 Database Development

Volumetric Surface object is created by the surface *(s)*, polygon *(p)*, line *(l)*, and point *(v)*. In representing spatiotemporal objects, time *(t)* is also involved to represent the period of changes and duration of changes. The changes on the surface comes from the points of the object that affects the lines that are joined together to build a triangle or polygon and surface.

The normalization of entities, attributes and relationships provides the logical model. The first level of normalization process is identifying the entity and attribute with relationship. Second level is identify dependency of the attribute with the entity. In this case, a new attribute will be created if required. Third level is to identify recursive attribute in the entity and come out with minimum redundancy of the attribute. Here, the management of the temporal element (time) is very important so that it can represent a movement without any redundancy issue. For purposes of managing the surface movement, we need to create a new entity surface movement. Surface movement entity will cater surfaces and time in the spatiotemporal data management. However, a complicative

issue with point entity could be raised if time is included in the entity as an attribute. For this purpose, point entity were evolved into temporal point entity which contains point data with temporal element. In both entities we create an index as identification in the entity. Here, we produced suitable logical data model, represented in the Entity Relationship Diagram (ERD) in Fig. 2.

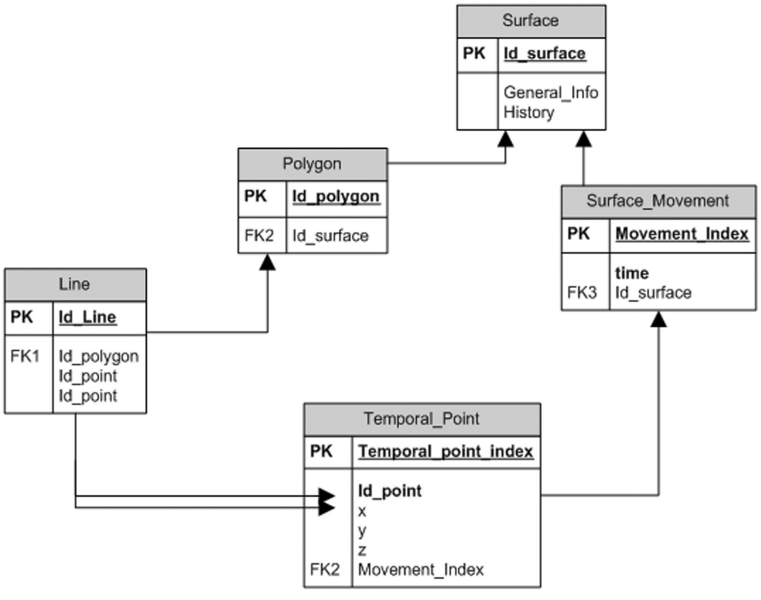


Fig. 2. Entity Relationship Diagram for VSMST Data Model

In the ERD for VSMST shown in Fig. 2, there are five entities that need to be created in the database to store and manage volumetric surface movement data. These are Surface_Movement, Surface, Polygon, Line and Temporal_Point. Every entity has relationships with related attribute. Table 2 shows the description of these entities. Table 3 shows the relationship of these entities with related entity.

4.2 Data Loading Process

In the data loading process, data will be loaded into the database from Triangulate Irregular Network (TIN). The data loading need to consider the format of the TIN structure and the database structure. From the TIN structure we captured the points, lines and triangles to be stored in the database, and we also considered the arrangement of the data to represent the correct surface. For loading of the data, identification of the data also needs to be indexed. Index of data currently is based on the feature identification with time. Every point has identification and their time as an attribute. The main principle of this database system, is when a point was not involved with movement, then it will not

Table 2. Description of the entities in the Logical Model

Entity	Description
Surface_Movement	Surface_Movement entities contain information about set of surfaces involved with movement. Attribute that will be stored in this entities are Movement_Index which is an indexing of the surface movement object, Id_surface which is surface involved with movement and also time of the occuring movement
Surface	Surface is an entity about the surface information. This is the place where this model can be extended with other information. In this entity we have Id_surface which is index of surface, General_Info and History which is an additional attribute based on user requirement
Polygon	Polygon is an entity that will store the information about the polygons on a particular surface. Thus, in this entity we have an Id_surface polygon and Id_surface
Line	Line stores information about construction of lines where the lines belongs to a particular surface. The attribute are are Id_line, id_point (start point), id_point (end point) and Id_polygon
Temporal_Point	Temporal_Point basically contains information about data which created the surface movement. So the attribute must be a coordinate of (x,y,z) and the temporal component. Temporal element is the same as time involved in the surface movement, so that we just include the Movement_Index as one of the entity. However, we have temporal_point_index to index the data which involved with the movement in the specific surface

Table 3. Description of Relationship in the ERD

Entity	Relationship	Entity
Surface_Movement	Has	Surface
Polygon	Member of	Surface
Line	Member of	Polygon
Line	Construct by	Temporal_Point
Temporal_Point	Movement	Surface_Movement

be stored. This will avoid data redundancy of the points in the database. The algorithm to perform this operation is given below.

1. Read data from the data source based on time (t)
2. While not end of file
3. read <id_vertex, x, y, z>
4. if <id_vertex, x, y, z> not in database

5. load the data
6. else if id_vertex on the database
7. if <x, y, z> = <x, y, z> in database
8. does not involve with changes, do not store the point
9. else store <id_vertex, x, y, z> with time (t)
10. end

4.3 Data Retrieval Process

Query process was designed to retrieve data from database by using area identification. A surface entity in the database has a Id_Surface with a start time and end time. Thus, results from this query will be a set of points which represents volumetric surface at the start time and end time. Below is the query algorithm that was used.

Data Retrieval Process:

1. *Get input from user; Area of the surface (A), Start Time (Ts) and End Time (Te)*
2. *Select Surface required based on the A and Time in between Ts and Te*
3. *Select Point based on the coordinate of area A and Time in between Ts and Te*
4. *Create surface based on the Time Start (Ts)*
5. *Calculate number of changes involved in the movement process*
6. *Identify list of vertex involved with changes based on time between Ts and Te*

This data will then be transformed into a file format which will be used to visualize the volumetric surface movement interactively by navigation of time. The file format uses data which were constructed from the start time given by the user. Number of changes in the selected area will be stored effective from the reformed start time. Then data (set of points) will follow subsequent changes and continue until the end of changes. Every change will only contain a set of data which were involved during the changes.

5 Data Visualization

In general, there are two steps in visualizing these changes. The first step is to establish a mapping from each point on the source object to some point at the targeted object. Once the correspondence is established, a sequence of intermediate models is created by interpolating corresponding point on the surface of the source objects to their position on the surface of targeted objects. Computing correspondences are done with respect to the minimal center distances. The established correspondence is one-to-one and thus, one vertex on source object

is paired with one vertex on targeted object. The interpolation part is solved by using parametric equation. The equation between two values used is:

$$Mv = P_{ti} + \{(P_{u+1} - P_{ti}) * l\}$$

Whereby l represents value that ranges from (t_s) to (t_e)
Mv represents the InBetween value
P_{ti} represents point in start time
P_{u+1} represents point in end time.

This formula needs to be applied to every parameter that will vary during the interpolation. For instance, in this simple application, 3 spatial dimensions system is used and those parameters would be x, y and z.

6 Results of Implementation

This section will discuss the result of implementation of the model with the prototype application. For the testing phase, conceptual data were used which is 16 points with two sets of movements, 12.5% percent of the data were involved

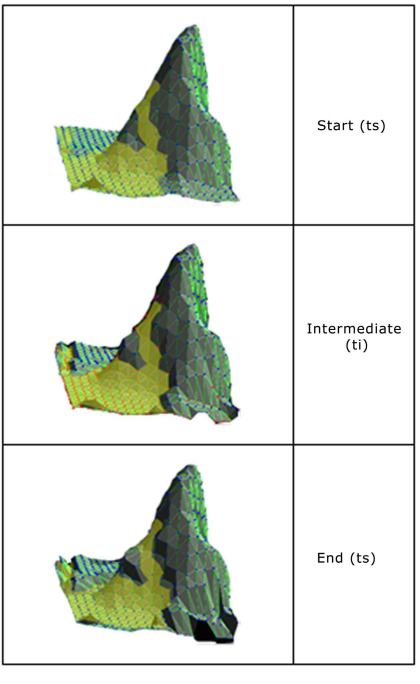


Fig. 3. Simulation of the Surface from Start Time (ts) to End Time (te)

Table 4. Redundancy test before and after loading process

Data Set	No. of Vertex Before Store	No. of Vertex After Store	Vertex Reduced
1	304	122	182
2	612	252	360
3	923	385	538
4	1212	492	720
5	1507	614	893
6	1809	729	1080
7	2123	863	1260
8	2454	966	1488
9	2732	1124	1608
10	3008	1235	1773

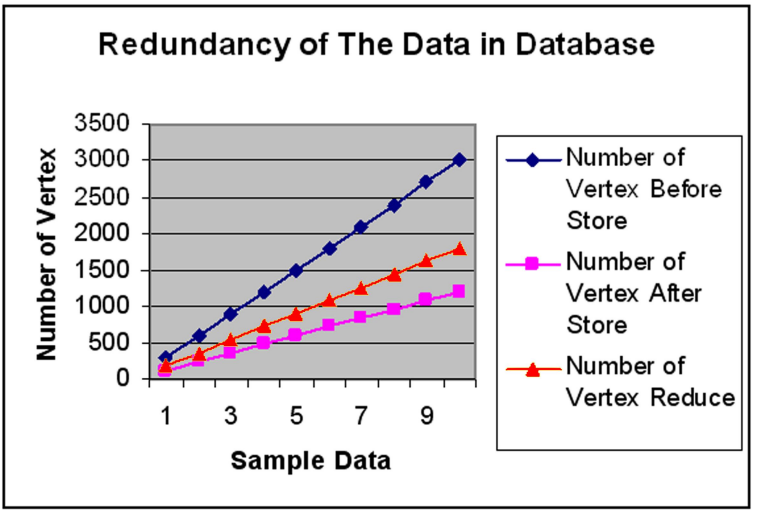


Fig. 4. Graph of the Data Redundancy before and after loading data into database

with movement. Before loading, there are 48 points in which only 20 remain after loading process. So, we were able to reduce the data by 58.3% by avoiding data redundancy. Figure 3 shows the simulation of data from the start time (ts) to end time (ts).

From the testing result of ten data samples, this model were able to reduce around 60% of the data. Overall the data has been sampled by assuming that 10% of the vertex points will be involved in the movement in the surface with three times of movement. However the reduction of the data is based on how many data is stored and how many vertex points involved with movement. Table 4 shows the result of the testing process and Figure 4 illustrates the graph result.

7 Conclusion and Future Works

According to [14], a successful VGIS query process should be able to support different user preferences in spatiotemporal scene queries, and do not have a fixed-metric approach where all users are considered equal. This requires a spatiotemporal database that can integrate series of data. The Volumetric Surface Movement Spatiotemporal Data Model can be an ideal solution to solve this kind of issue. We have developed a prototype having the ability to reflect surface changes of a volumetric object. We have also included the ability to efficiently compress data not involved in the change. This model has shown to be able to reduce data redundancy by 60%.

As a conclusion, our major contribution in this paper is the implementation of the Volumetric Surface Movement Spatiotemporal Data Model by using the relational database model in a prototype version. The proposed data model can be considered in the development of the related application. In the visualization tool we have used a data format which were generated from the Spatiotemporal Data Retrieval Processor to load into the Volumetric Surface Movement Simulator. This aids the use in visualizing the result.

Acknowledgments. This research has been sponsored by the Ministry of Science, Technology and Innovation (MOSTI), Malaysia. Research has been managed by Research Management Centre, University Technology Malaysia under research grant 79102. Special thanks to the Institute of Advanced Technology for the advice and guidance given during research.

References

1. Yuan, M., Mark, D.M., Egenhofer, M.J., Peuquet, D.J.: Extensions to Geographic Representations. In: McMaster, R.B., Usery, E.L. (eds.) A Research Agenda for Geographic Information Science, Boca Raton, Florida, ch. 5, pp. 129–156. CRC Press, Boca Raton (2004)
2. Nadi, S., Delavar, M.R.: Spatio-Temporal Modeling of Dynamic Phenomena in GIS. In: Proceeding of ScanGIS 2003, pp. 215–225 (2003)
3. Rahim, M.S.M, Shariff, A.R.M, Mansor, S., Mahmud, A.R.: A Review on Spatiotemporal Data Model for Managing Data Movement in Geographical Information Systems (GIS). Journal of Information Technology, FSKSM, UTM 1(9), 21–32 (2005)
4. Rahim, M.S.M, Shariff, A.R.M, Mansor, S., Mahmud, A.R.: Volumetric Spatiotemporal Data Model, Lecture Note in Geoinformation and Cartography, pp. 547–556. Springer, Heidelberg
5. Narciso, F.E.: A Spatiotemporal Data Model for Incorporating Time in Geographical Information Systems (GEN-STGIS), Doctor of Philosophy in Computer Science and Engineering, University of South Florida (April 1999)
6. Ale, R., Wolfgang, K.: Cell Tuple Based Spatio-Temporal Data Model: An Object Oriented Approach, Geographic Information Systems, Association of Computing Machinery (ACM), 2-6 November 1999, pp. 20–25. ACM Press, New York (1999)

7. Moris, K., Hill, D., Moore, A.: Mapping The Environment Through Three-Dimensional Space and Time, Pergamon, Computers. Environment and Urban Systems 24, 435–450 (2000)
8. Donggen, W., Tao, C.: A Spatio-temporal Data Model for ActivityBased Transport Demand Modeling. International Journal of Geographical Information Science 15, 561–585 (2001)
9. Philip, J.U.: A Spatiotemporal Data Model for Zoning, Masters of Science Thesis, Department of Geography, Brigham Young University (2001)
10. Bonan, L.,Guoray, C.: A General Object-Oriented Spatial Temporal Data Model, Symposium on Geospatial Theory, Processing and Applications, Ottawa (March 2002)
11. Elena, C., Michela, B., Elisa, B., Giovanna, G.: A Multigranular Spatiotemporal Data Model. In: GIS'03, New Orleans, Louisiana, USA, November 7-8, 2003, pp. 94–101 (2003)
12. Yanfen, L.: A Feature-Based Temporal Representation and Its Implementation with Object-Relational Schema for Base Geographic Data in Object-Based Form, UCGIS Assembly (2004)
13. Roddick, J.F., Egenhofer, M.J, Hoel, E., Papadias, D.: Spatial, Temporal, and Spatiotemporal Databases. In: SIGMOD Record Web, vol. 33(2) (2004)
14. Mountrakis, G., Agouris, P., Stefanidis, A.: Similarity Learning in GIS: An Overview of Definitions, Prerequisites and Challenges, Spatial Databases Technologies, Techniques and Trends, pp. 295–321. Idea Group Publishing, USA (2005)
15. Anders, W.: Book Review: Next Generation of Geospatial Information from Digital Image Analysis to Spatiotemporal Databases by Peggy Agouuris and Arie Croitoru. Transaction in GIS 10(5), 773–775 (2006)

Query Distributed Ontology over Grid Environment

Ngot Phu Bui, SeungGwan Lee, and TaeChoong Chung

Department of Computer Engineering, Kyung Hee University, Republic of Korea
bpngot@iislab.khu.ac.kr,
{leesg, tcchung}@khu.ac.kr

Abstract. On the Semantic Web, query answering will require data coming from many different ontologies, and information processing is not possible without having a framework that discovers their overlaps, evolved contents, and availabilities. Semantic searching on predefined number of ontologies may give us incomplete, unclear, and even meaningless result. Hence the development of a system to exploit the relationship of semantic sharing data from various sources for searching process is crucial to the success of the Semantic Web. We propose Distributed Ontology Framework on Grid environment (DOFG) and a query analyzing method in distributed ontology environment. DOFG employs Grid computing technique to build an ontology virtual organization for managing meta-data and utilizing the computing power of many sites when processing query. Our implementation on some real-world domains shows the fitness of DOFG on large sharing semantic data mediums and the high feasibility of this framework to be a generic architecture for numerous semantic grid applications.

Keywords: Semantic Web – Ontology – Query answering – Semantic Grid.

1 Introduction

The World Wide Web in its current form is a dramatic success with the growing number of users and information sources. It currently contains around 3 billion static documents which are accessed by over 500 million users internationally. However, the continued rapid growth in information volume makes it increasingly difficult to find, organize, access and maintain the information required by users. Tim Berners-Lee [1], the inventor of the WWW, refers to the future of current Web as Semantic Web that provides enhanced information access based on the exploitation of machine-processable metadata. This metadata defines what the documents are about in well-defined meaning and machine-readable way. Therefore, the Semantic Web will be able to support various automated services based on these representations. These presentations are regarded as the main factor to finding a way out of the growing problems of traversing the expanding web space, where most web resources can currently only be found through syntactic matches (e.g., keyword search).

Ontologies have shown to be a key enabling technology for the Semantic Web. It offers a way to cope with heterogeneous representations of web resources by providing a formal conceptualization of particular domain that is shared by a group of

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, pp. 140–153, 2007.
© Springer-Verlag Berlin Heidelberg 2007

people. The domain model implicit in an ontology can be taken as a unifying structure for giving information a common representation and semantics.

The Semantic Web and ontology are often developed by a particular organization. An ontology is built so that it can serve for a specific domain. In other words, ontology is domain-dependent. A major problem is that current infrastructure of the Internet and web browsers have no general way for us to combine all related information from many web resources. The following example will illustrate a clear need to have such system that weaves together information on the Semantic Web.

Example 1.1: Suppose there are semantic sites about academic document, university, and personal information domains respectively and you want to find out the some research papers, the author of them, and her email or phone number. You know that these papers are related to "*X*" field and that the author teaches at University "*Y*". You also know that her research interest is "*Z*".

You would have trouble finding the above information currently because it is not contained within a single semantic website. Our system, Distributed Ontology Framework on Grid environment referred as DOFG, allows ontology metadata to be shared across network so that we can obtain the answer quickly for the above circumstance. As we are in a large-scale and heterogeneous P2P network like Internet, it is not feasible for a server to get data from all sources and then do the searching on its own. Thus, we propose a framework with the help of Grid computing ([2], [3]) technology to enable flexible control of ontology metadata from different semantic sites. Taking full advantage of collaboration among peers when processing query will enhance performance of system.

Query processing in the example above requires data access to three ontologies. With the constraint "*paper related to "X" field*" and academic document ontology data, we can retrieve more information about papers having such criterion and so on for other constraints in a query. However, the way to identify which ontologies are needed and to mine appropriate information from these ontologies for answering query is rather complex and critical problem.

The remainder of the paper is organized as follows: Section 2 introduces related work about P2P network applications and metadata query processing on those systems. Section 3 presents the overall and node architecture of DOFG. Section 4 describes the algorithms used to create sub-queries, routing and processing those sub-queries on relevant peers. We discuss our implementation in Section 5. Finally, we conclude our research in Section 6.

2 Related Work

Through many recent similar researches about P2P network applications, we see that there are two kinds of P2P network systems, pure P2P network and schema-based P2P network. The first form, with the characteristic of full decentralization, has the advantage for avoiding bottleneck problem when many peers in the network have access to the centralized points, however, the "*query flooding*" to many unnecessary

peer nodes when processing requests from user is the drawback of this solution. The absence of global authority server in [14] for being convenient when joining or leaving action of node happening but based on [2] and [4], we can handle this issue well when setting up Index Service for Virtual Organization (VO). In contrast to ObjectGlobe [19] that every joining/leaving action leads to adding or removing corresponding metadata in the repository, we use "*active/inactive*" mechanism to mark the metadata. Every peer becomes a schema-data advertising server in [5] may increase redundancy of data in the system and be less efficient for a large heterogeneous network (e.g. Internet). Pastry framework [17] is deployed in [6] requires the peer equality and also peer source of metadata like [5]. DHT-based system ([13], [20]) has been shown to be capable of integrating various information sources in query mechanism answering to users. However, the data relation between nodes in this framework is tight so it will be difficult for synchronizing when peer joining/leaving actions happen. To better avoid overwhelming the network with query, DOFG has a layer that manages and processes metadata of all peers. More concretely, this layer is complementary to Globus MDS4 [4] and represented by a group of interacted Ontology Index Servers. These servers contain mapping information between ontologies and nodes in the grid environment.

The second form of P2P network is often designed with a central server or group of servers that manage all the peers in the system. An extension based on Sesame for querying distributed RDF repositories in [11] by using RDF-graph index structure to optimize query process is nearly corresponding to our approach but peers' interactions are still absent. Piazza [18] is also a schema-based P2P system which exploits data format and document structure mapping among small sites or just pair of nodes in the network for answering query. Edutella super-peers [10] and its successors ([15], [16]) employ routing indices that explicitly acknowledge the semantic heterogeneity of schema-based P2P networks and include peers' schema-data storing. Our approach is similar to these three studies in the aspect that we also use the scenario that stores metadata of all ontologies for routing query between nodes in network. Nevertheless, the lack exploiting collaboration between peers of these above systems is the main difference as comparing with ours. We use the hybrid architecture to take advantage of computing power of all peers. One node can require specific data from another and use as a filter for its local query process.

3 Architecture

Figure 1 shows the overview architecture of DOFG system. It consists of three following main components: Routing Agent, Ontology Index Service (OIS), and Data Processing Unit (DPU). Routing Agent plays a role as Web server to many users. Web users submit the RDF [7] query in SeRQL [8] language to Routing Agent. The essential task of this component is to determine which OIS is in lowest load state and send the query to it. Then, Ontology Index Service decomposes receiving query into atomic or complex sub-queries [6]. Query routing is performed in this component for finding the relevant peers by taking matching processes in the ontology metadata, the RDFS database in OIS. Answering to each sub-query is done by searching on local

RDF instance database of corresponding DPUs. In addition to part-query answering, each DPU also takes the responsibility of aggregating partial RDF data searching result. Finally, complete answer will be returned by some DPUs.

In this architecture, a small subset of nodes, called OISes, takes over specific functions for peer leaving/joining, query decomposing, sub-querie routing, and metadata replication. In DOFG, when a site leaves the Ontology Virtual Organization (OVO), its metadata is marked as in "*inactive*" state. It means that its metadata is not available for searching process. Furthermore,

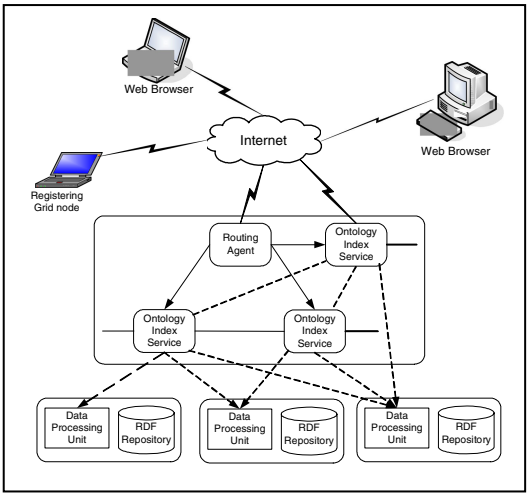


Fig. 1. The DOFG architecture

DOFG also has mechanism that allows one or more OISes periodically detect the aliveness [21] of peer and then forward any leave to the rest OISes. On the other hand, when joining, a site is asked for its membership identifier (MID). If this identifier is available in the MID directory of Index Service, this site has been a member of OVO and vice versa. A node will advertise all its schema-data to Index Services and be assigned a unique MID when first registering in OVO. With a rejoined node, Index Service will ask for a version of ontology schemas and update the database incase the new version of node's schema is detected. Currently, DOFG just requires full new version reregistration in Index Service if change in node schema data has happened. If node's schema data is detected to be unchanged version, we mark its metadata as "*active*" state.

All the OISes operate in the same way when routing sub-queries so their metadata repositories must be similar. New node joining, schema updating or rejoining of old node with the changing of data will lead to data updating and replication among OISes. With the purpose to balance the amount of work as much as possible, we also have a simple method that can determine the load information of a specific OIS in predefined period of time T_0 through formula:

$$R = N/T_0 \tag{1}$$

where N is number of queries submitted to that OIS during T_0. Load information R will be sent to Routing Agent periodically so that Routing Agent has a flexible decision to route user's query to appropriate OIS.

3.1 Node Architecture

In this section, we briefly show block buildings that contribute to a node of our system. From figure 1, we see that DOFG system includes many DPUs and OISes.

These nodes have the same structure presented in figure 2. Local Ontology Index is index service often used to maintain a registry of interesting services and resources in DPU nodes. Nevertheless, the index service in OIS, Broker Ontology Index, has some differences in function. It serves as the top-level index service for the ontology VO, as all local indexes will register with it to advertise their ontologies. The broker maintains a registry of interesting resources for the entire VO. The "*resources*" of DOFG are not CPUs nor storage capacity, but RDF/S data files. In our system, DPU index service manages local schema ontology files, provides web services about aliveness and MID information. On its turn, OIS index service plays a role as upholder of all ontology file resources in VO and supplies web services about query preprocessing, schema advertising, peer registering, schema updating, load information, and sub-queries routing. These services can be called by other web service in OIS or DPU nodes.

Aggregator Service is responsible for aggregating and joining query sub-results between DPUs. Aggregator in an OIS is used for performing schema data synchronizing process when receiving update information from another OIS. Looking up on schema and instance repositories for answering user's query is performed by *Query Service*.

This study focuses on sharing and querying RDF data so a library for storing and manipulating on that data is indispensable. Fortunately, we will not need to implement this library ourselves, as Sesame [9] is a generic framework for storing and processing RDF data. In DOFG, we use Repository API and RDF I/O (Rio) of Sesame to handle those tasks. These APIs provide various methods for uploading data files, querying, and extracting data.

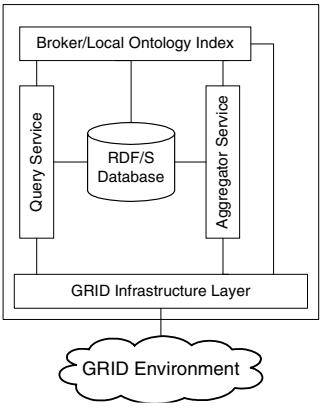


Fig. 2. Node architecture

3.2 Web Service in DOFG

Web services are platform-independent and language-independent since they use standard XML languages and HTTP protocol for transmitting request/response messages.

These are major advantages that enable us to use web services for building Internet-scale applications. Web service is the fundamental mean for interaction among peers in DOFG. Every peer can be a web service provider or web service consumer. For example, fresh DPU A, service consumer, invokes the schema advertising web service of one OIS, service provider, through the URI *https://163.180.116.133/ ois_service/SchemaAdvertisisng*. The web service architecture includes four main parts: service processes, service description, service innovation and transport. Discovery of service processes allows us to locate one particular service from a collection of web services. What operations that one web service can support and how to invoke it are undertaken by service description and described in WSDL [22] format. Invoking a web service involves passing messages between client and server. In our system, SOAP [23] is used as service invocation

language to specify how we should format the requests to the server, and how the server should format its responses.

3.3 Ontology Virtual Organization

The purpose of DOFG framework is to utilize semantic data from various semantic data resources on the Internet. Using Grid technology, we can achieve greater performance and throughput by pooling together resources from different organizations. Resources from several different organizations are dynamically pooled in Virtual Organization (VO) [2], OVO in our case, to solve specific problem. In the design of DOFG, we follow the nature of Grid computing. It means that the membership of a grid node is dynamic: resources in OVO often come and go. For example, in figure 1, Registering Grid Node is the node that wants to join in OVO.

4 Query Decomposing, Routing and Processing in DOFG

Given a set of semantic data sites, their indices of metadata, and a query from a user, the key issue we have to deal is how to process the queries efficiently and obtain semantically correct answers. Our study focuses mainly on reformulating the query and determining which part of the overall query as well as filter has to be sent to which appropriate site. As described in the second section, P2P networks can be divided into two categories: pure P2P networks and schema-based P2P network. In this section, we show the deficiencies of traditional query processing in both categories. Then we propose our hybrid approach on part of query routing and processing in a dynamic and scalable schema-based P2P networks like semantic grid.

4.1 RDF Data

The resource description framework (RDF) is a recent W3C recommendation designed to standardize the definition and use of metadata description of web-based resources. It is a language for describing semantic metadata about resources in the Web by basing on the idea of identifying things using Uniform Resource Indicators (URIs). RDF provides a solid base to establish enterprise semantic applications and has implied a significant leverage of the Semantic Web. The basic building block in RDF is a subject-predicate-object (<s, p, o>) triple. RDF schema takes a step further into richer representation formalism and introduces basic ontological modeling primitives into a web-based context by letting the developers define particular vocabulary for RDF data. In other words, the RDFS mechanism provides a basic type system for RDF models. This type system uses some predefined terms, such as `Class`, `subPropertyOf`, and `subClassOf`. RDF objects can be defined as instances of one or more classes using `type` property. The `subClassOf` property allows the developer to specify the hierarchical organization of classes.

Ontology is the data structure used to express the semantic information on the Semantic Web. RDF/S, DAML+OIL, OWL are the knowledge representation languages often used to represent ontology.

Currently, our system can handle the searching for RDF/S ontology-based. It means we supposed that all the semantic sites registering to OVO use RDF/S for representing their data. Figure 3 represents ontology snippet whose domain is related to *academic document*. For evaluating our system, we used Protégé [24] to create some ontologies along with their instance data and place them on separated sites. For example: the *academic document ontology* represents 81 classes about many kinds of document, the properties as well as relationships of these classes.

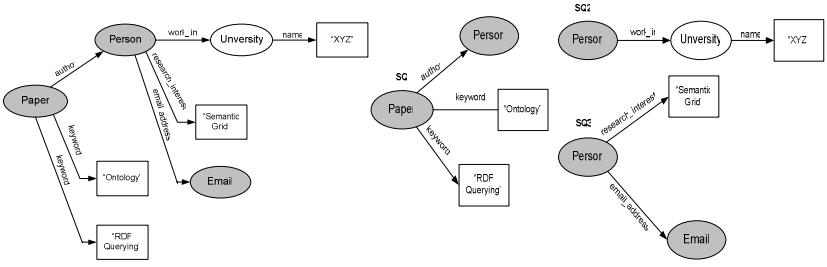
Fig. 3. Document Ontology Snippet

4.2 Query Decomposing and Matching

In this paper we supposed the queries submitted to the Index Services are written in Sesame RDF Query Language (SeRQL). We decompose query into atomic queries with a format (*subject, predicate, object*) or sub-queries made from atomic ones. Atomic query is a triple pattern in which the *subject*, *predicate* (also called *property*), and *object* can be each either a variable or exact value There are eight possible atomic triple queries for exact matches (see atomic triple patterns in [12]). One of the most prominent parts of SeRQL is path expression. Path expressions are expressions that match specific path through RDF query. Each and every path can be constructed using a set of basic path expressions. In our case, we assumed that basic path expression is atomic triple and path expression with length [8] greater than one is sub-queries. The following example will illustrate decomposing process.

Example 4.1: Suppose that we want to ask for articles that relate to "Ontology" and "RDF Querying". The path expression of corresponding query would be described in figure 4a. Gray ovals represent for variables or information that we want to know.

The decomposing process bases much on the query path expression matching to ontology RDF graphs. In the example above, we decompose complete query Q into three sub-queries (SQ1, SQ2, SQ3) demonstrated in figure 4b. This implies that there are three ontologies containing appropriate data for answering these sub-queries. In our approach, we assume that Index Service contains RDF schema graph models of all sites. Schema is a collection of triples, each consisting of a subject, a predicate and an object. In a triple, subject can be a URI reference or a blank node, predicate as an RDF URI reference, and object as an RDF URI reference or a literal or a blank node. In other words, RDF Schema Graph is a graph $G(V, E)$ in which V is a set of nodes representing for subjects and objects and E is a set of edges representing for



a. Full Query Q **b.** Sub-queries

Fig. 4. Query decomposing

predicates. We carry out path expression matching by projecting query triples on RDF schema graph. This process will help us to discover which part of the query subsumed by which RDF schema.

Definition 1: Let $G = (G_1, G_2, ..., G_n)$ be available ontology RDF schema graphs. RDF query graph G' is decomposed into a set of subgraphs $(G_1', G_2', ..., G_1')$ if and only if a projection exists such that for each $G_{i(i<k+1)}'$, there is at least one $G_{j(j<n+1)}$ for projecting G_i' into G_j

A sub-graph G_i' is able to be projected into graph G_j if and only if each node/predicate of G_i' is projected on each node/predicate of G_j. Nodes or predicates projection meets when they have the same value. Figure 4 shows an original query graph and its corresponding sub-queries after decomposing. In order to find which part of query relevant to which ontology, we have to carry out the matching process between a specific query and metadata ontology on Index Service. Ontology matching is also research topic which has been being studied by many researchers. Some systems such as GLUE [25] provides a more generic form of matching that permits flexibility in selecting the similarity measure. Bipartite graphs [26], as an intermediate model for RDF, used for Graph Matching for Ontologies [27] or matrix representation of nodes from an RDF labeled directed graph used for OWL ontology alignment in [28] are useful approaches for measuring the structural and semantic similarity between ontologies. In our system, we use different approach for decomposing and matching a query to ontology schema. In order to decompose the query, we base on SET (Similar Entities Table).

Definition 2: SET is a table of common entities (concepts) and corresponding ontologies that contain those entities.

$$SET = \{(E_1, (O_{11}, O_{21}, ..., O_{r_1 1})), (E_2, (O_{12}, O_{22}, ..., O_{r_2 2})), ..., (E_l, (O_{1l}, O_{2l}, ..., O_{r_l l}))\}, \quad E_i \in \bigcap_{j=1}^{r_i} O_{ji}$$

Entities are the general concepts expressed by classes in ontologies. SET content is a set of pairs having $(E_i, (O_{1i}, O_{2i}, ..., O_{r_i i}))$ format. This format indicates the existence of entity E_i in ontologies $O_{1i}, O_{2i}, ..., O_{r_i i}$. In definition 2, we assume number of common

entities in SET is *l*. When decomposing query, we have to determine which its entities are in SET table and this investigation will give us a subset of common entities belonging to that query. At this time, we know some candidate ontologies used for query answering. We can see that with a query, there may be more than one decomposed solutions. And, we will find all possible decomposing for a specific query from the above candidate ontologies. When processing at a node in query graph (a common entity), we store all possible matching between part of query graph and corresponding ontologies containing that common entity. Figure 5 describes pseudo code of decomposing query algorithm.

```
Procedure Decompose_Query()
Input    : query graph G'
Output:  possible decomposed solutions
//Possible decomposed solutions, partial matching
PDS = φ ;  PM = φ
//Query entities
QE = Breadth_First_Search(G')
//Common entities in query
SE = QE ∩ SET
foreach E_i ∈ SE do
   //Ontology subset OS contains E_i
   OS_i = {O_1i, O_2i, ..., O_ri}, E_i ∈ ⋂_{j=1}^{r_i} O_ji
   //Find all matching M for sub-query with root E_i
   M(E_i) = {OS_1i, OS_2i, ..., OS_ki}, OS_ji(j<=k) ⊂ OS_i
   PM = PM ∪ M(E_i)
endfor
//Combination matching set
CMS = Create_Combination(PM)
foreach CM_i ∈ CMS do
  //Marked entities set of CM_i
  MES = Entity(CM_i)
  PM' = φ
  foreach E_i ∈ QE \ MES do
    //Find matching for uncommon entities
    PM' = PM' ∪ M'(E_i)
  endfor
  //Adding one possible decomposed solution
  PDS = PDS ∪ Solution(CM_i, PM')
  Optimize(PDS)
endfor
endprocedure
```

Fig. 5. Decompose query pseudo code

One decomposed solution contains an instance of $(G_1^{'}, G_2^{'}, ..., G_k^{'})$, subset of ontology and mapping information between sub-graph $G_i^{'}$ and ontology O_j. It is clear that we may have more than one distinct subset of ontology that can answer for one instance of $(G_1^{'}, G_2^{'}, ..., G_k^{'})$, Thus, we have to refine all possible decomposed solutions by determining which ontologies can answer a specific sub-graph $G_i^{'}$. We solve this problem through optimized function in above algorithm. Let $p_1, p_2, ..., p_k$ be number of ontologies used for answering subgraphs $(G_1^{'}, G_2^{'}, ..., G_k^{'})$, respectively. As a result, we have a formal format for one arbitrary decomposed solution.

$$(G_1^{'}, (O_{11}, O_{12}, ..., O_{1p_1})), (G_2^{'}, (O_{21}, O_{22}, ..., O_{2p_2})), ..., (G_k^{'}, (O_{k1}, O_{k2}, ..., O_{kp_k}))$$

4.3 Query Routing in P2P Networks

All query approaches are constantly concerned with providing improved search functionalities in P2P network. As schemas describe the content stored at peers, query can be more precise and flexible by using these schemas. Nevertheless, if we used simple broadcast topologies for these networks, queries are broadcast to all peers and consume the network bandwidth and computing power at each peer. The Pastry P2P used in [6] provides a scalable, decentralized and self-organizing framework for routing messages from one peer to another. However, the flooding of sub-queries in the network is inevitable. The broadcasting of RDF schemas from one node to other peer nodes in SON [5] enables every node to be capable of creating query plan which holds the information about a set of peers that need to be conducted for executing the query but this approach requires every peer contain the whole basic data of all sources. With the same method, propagating information [14] to every node in the network whenever a new node joins may seem limiting for the network's scalability.

Processing query in schema-based P2P networks is much more efficient as the metadata indices guides the searching to the appropriate peers. The necessary peers in the networks communicate each other and flooding the network with requests is avoided. Obviously we have to investigate more efficient approaches which query only those peers that are indeed able to understand and answer the query. Queries can be distributed relying on the metadata routing indices. These routing indices do not rely upon on a single schema but can contain about arbitrary schemas used in the network. Mediator SAIL [11] takes the responsibility of distributing sub-queries to suitable nodes but the model for utilizing the peers' interaction for query processing and peer joining is still absent.

Super-peer-based routing topologies ([10], [15], [16]) have been shown to be better scalable than the metadata broadcast-based network and do provide perfect support for heterogeneous schema-based networks, which support different metadata schemas and ontologies. Super-peers in Edutella network are arranged in the HyperCuP topology that enables to organize super-peers of a P2P network into a recursive graph structure call a hypercube. This solution makes the system be easy for integrating the new super-peer and avoiding the central maintenance. Furthermore, it enables efficient and non-redundant query broadcast due to super-peer/peer (SP/P) and super-peer/super-peer (SP/SP) routing indices. The SP/P indices contain information about metadata usage at each peer and the SP/SP ones are the summaries from all super-peers local SP/P indices. Query routing in this approach includes two routing

scenarios: super-peer to super-peer and super-peer to peer. The similarity of our routing algorithm to the above one is that we also use special a small subset of peers which store metadata of all peers. Consistency of local metadata among super-peers (OIS) in DOFG is assured by broadcasting the peers joining/leaving and peers data updating events from one super-peer to others. Thus, we need not the routing between those peers when answering user request. Our strategy can be described as the following three steps.

- Routing Agent sends the query to the most appropriate OIS node and then OIS will analyze query graph into sub-queries.
- With a decomposing solution (subsection 4.2), we route each sub-query to suitable node along with an aggregating peer MID identifier. In case there is more than one peer which can answer for routed sub-query. We will randomly choose one peer that takes responsibility for uniting partial results.
- Aggregate Service on each receiving sub-query peer submits sub-query result to node that has MID similar to aggregating peer MID for data union or joining.

Example 4.2: Examining the previous example, we have three sub-queries SQ_1, SQ_2, SQ_3 with decomposing solution on three subsets over five ontologies $\{O_1\}$, $\{O_2, O_3\}$, $\{O_4, O_5\}$ separately located at peers A, B, C, D, E. According the routing strategy, we route SQ_1 to A with no aggregating peer MID; SQ_2 to B, C with A's MID and B's MID; SQ_3 to D, E with A's MID and D's MID respectively.

4.4 Query processing

Query processing in DOFG is responsible for generating and combining partial results on relevant distributed peers to obtain a complete answer. Figure 5 illustrates an example of how sub-queries shown in example 4.1 can be executed over DOFG peers. From example 2, we know that OIS sends sub-queries and as well as other related information to relevant peers. In order to get partial result, a peer receiving a sub-query will perform the selection and projection operations with corresponding filter f_i (represented by *where clause*). Peer A generates partial result $SQ_1@P_A$ for sub-query SQ_1, peer B and C with $SQ_2@P_B$ and $SQ_2@P_C$ for SQ_2, and peer D and E with $SQ_3@P_D$ and $SQ_3@P_E$ for SQ_3. Union operations are executed at peer B and C to get a complete result for sub-queries SQ_2 and SQ_3. At that time, we have all partial results of three sub-queries. We can get the complete answer for the query Q through the performing joining operation of peer A with input $SQ_1@P_A$ $SQ_2@P_{B,C}$, $SQ_3@P_{D,E}$ and joining condition on entity Person.

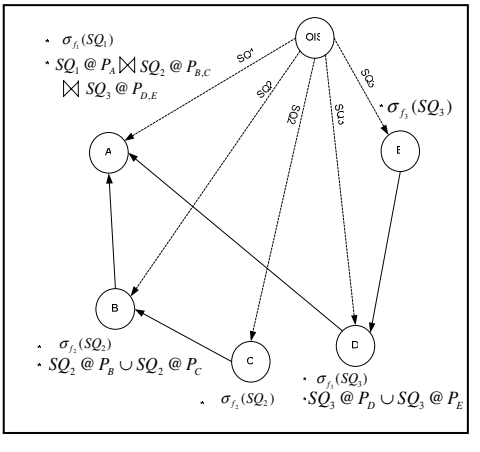


Fig. 6. Query processing

5 Implementation

We implement DOFG framework using Java on Linux environments. The principal technical features of DOFG are highlighted as below:

- The base development and deployment platform are Globus Toolkit 4.0.1 (GT4). All services are built based on the Open Grid Service Architecture (OGSA) standard.
- We used Protégé 3.1 to build ontologies and Sesame 2.0 to store, read, process RDF(S) data. For simulation, we use this tool to create three ontologies related to academic document, university, and personal information domains.

Currently, we tested our system on LAN environment with the speed of 100.0 Mbps. Table 1 describes physical nodes for overall system.

Table 1. Physical nodes in DOFG

Attr / Node type	Quantity	Operating System	Memory	CPU
Routing Agent	1	Fedora core 4.0	512MB	AMD 1 GHz
OIS	1	RedHat 9.0	512MB	PIV 2.4 GHz
DPU	3	RedHat 9.0 / Fedora core 4.0	512MB	PIV 2.8 GHz

We carry out our implementation using three ontologies. Each is deployed on one DPU respectively.

Table 2. Ontology information

	Document (O_1)	**University (O_2)**	**Personal (O_3)**
Class	81	32	19
Relationship	47	62	52
Intersection	12 ($\cap O_2$)	5 ($\cap O_3$)	6 ($\cap O_1$)

Table 3. Query used in the evaluation

No	Query
1	$\{C_1\}\ p_1\ \{C_2\}$ $p_2\ \{C_3\}$
2	$\{C_1\}\ p_1\ \{C_2\}\ p_2\ \{C_3\}$ $p_3\ \{C_4\}$

In order to evaluate the performance of the system, we create queries that need the data from two or more ontologies for answering. In our evaluation, we use two kind of query: query with two triples and query with three triples. Each triple is answered by one ontology. We create instance data file for each ontology from 10 MB to 80 MB stored in Sesame PostgreSql. Figure 7 shows the performance measurements for query answering process. The result obtained in the

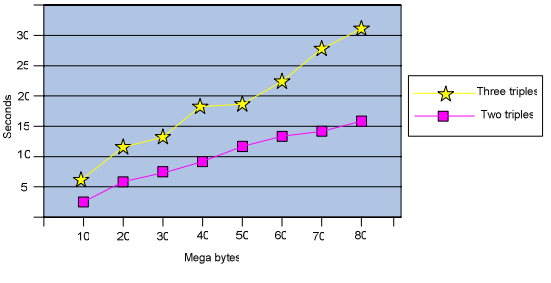


Fig. 7. Query answering performance

query tests reflects time for answering query with two triples is nearly twice less than that of three-triple query since the former is decomposed by two sub-queries. However, the latter is decomposed into three sub-queries and its final result available after two times for aggregating partial results.

6 Conclusion and Future Work

In this paper we have discussed an infrastructure for querying ontologies in a distributed environment. We see a framework allowing collaboration among various information sources as key to enabling the success of semantic data query. Our approach focuses on using Grid technology to build a system that flexibly manages ontologies from heterogeneous sources and proposing an analyzed and routing query mechanisms on this system. Exploring and combining distributed ontologies from many different sites on large scale network are complex problems. These works requires the assistance through replication, updating, and advertising ontologies operations. We are developing grid services to tackle these operations.

In order to answer a query, we split a query into sub-queries and route them to the appropriate peers. Decomposing query algorithm gives us all possible decomposed solutions for a query based on ontology meta-data. Moreover, we have a mechanism to obtain the complete result for a query by aggregating partial results from peers through joining and union actions. This will help to reduce both the communication and processing costs.

To match triples we currently use the exactly terminologies for criteria of matching so our result will have limitations. In real world, there may be one concept described by many terms or semantic similarity exists among different terms or difference name entities share one literal. Our future work will be towards using disambiguating process to have better result in querying.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001)
2. Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organization. International J.Supercomputer Applications 15(3) (2001)
3. Foster, I.: What is the Grid? A Three points checklists. Grid Today (July 20, 2002)
4. The Globus Alliance.The Monitoring and Discovery System, http://www.globus.org/toolkit/docs/4.0/info/key-index.html
5. Kokkinidis, G., Christophides, V.: Semantic Query Routing and Processing in P2P Database Systems: The ICS-FORTH SQPeer Middleware. In: Lindner, W., Mesiti, M., Türker, C., Tzitzikas, Y., Vakali, A.I. (eds.) EDBT 2004. LNCS, vol. 3268, pp. 486–495. Springer, Heidelberg (2004)
6. Ranger, D., Cloutier, J-F.: Scalable Peer-to-Peer RDF Query Algorithm. In: Ngu, A.H.H., Kitsuregawa, M., Neuhold, E.J., Chung, J.-Y., Sheng, Q.Z. (eds.) WISE 2005. LNCS, vol. 3806, pp. 486–495. Springer, Heidelberg (2005)
7. Brickley, D., McBride, B. (eds.): RDF Schema, http://www.w3.org/TR/rdf-schema/
8. Sesame. The SeRQL query language, http://openrdf.org/doc/sesame/users/ch06.html
9. Broekstra, J., Kampman, A., Harmelen, F-V.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 54–68. Springer, Heidelberg (2002)

10. Neidl, W., Wolf, B., Siberski, W., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmer, M., Risch, T.: EDUTELLA: P2P Networking Infrastructure Based on RDF. In: 11th World Wide Web Conference (2002)
11. Stuckenschmidt, H., Vdovjak, R., Broekstra, J.: Index Structures and Algorithms for Queryign Distributed RDF Repositories. WWW2004, New York, USA (May 2004)
12. Cai, M., Frank, M.: RDFPeers: A Scalable Distributed RDF Repository based on A Structure Peer-to-Peer Network. In: Proceeding of the 13th International World Wide Web Conference (WWW), New York (2004)
13. Heine, F., Hovestadt, M., Kao, O.: Processing Complex RDF Queries over P2P Networks. In: 14th Information Retrieval in Peer-to-Peer Networks P2PIR2005, November 4th, Bremen, Germany (2005)
14. Galanis, L., Wang, Y., Jeffery, S.R., Dewitt, D.J.: Processing Query in a Large Peer-to-Peer System. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, Springer, Heidelberg (2003)
15. Nejfl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M., Brunkhorst, I., Löser, A.: Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks. In: Proceeding of the 12th International World Wide Web Conference (WWW), Budapest, Hungary (2003)
16. Brunkhorst, I., Dhraief, H., Kemper, A., Nejdl, W., Wienser, C.: Distributed Queries and Query Optimization in Schema-Based P2P-Systems. In: International Workshop on Databases, Information Systems and Peer-to-Peer Computing, September, Berlin, Germany (2003)
17. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In: IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pp. 329-350 (November 2001)
18. Halevy, A.Y., Ives, Z.G., Mork, P., Tatarinov, I.: Piazza: Data management infrastructure for semantic web applications. In: Proc. of the 12th Intl. World Wide Web Conf., (2003)
19. Braumandl, R., Keidl, M., Kemper, A., Kossmann, D., Kreutz, A., Prols, S., Seltzsam, S., Stocker, K.: ObjectGlobe: Ubiquitous Query Processing on the Internet. VLDB Journal, 48–71 (2001)
20. Heine, F., Hovestadt, M., Kao, O.: Towards Ontology-Driven P2P Grid Resource Discovery. In: proceeding of the Fifth IEEE/ACM International Workshop on Grid Computing (2004)
21. Iamnitchi, A., Foster, I.: On Fully Decentralized Resource Discovery in Grid Environments. In: Proceeding of the second IEEE/ACM International Workshop on Grid Computing, Denver (November 2001)
22. Web Service Description Language (WSDL) (2001), http://www.w3.org/TR/wsdl
23. Simple Object Access Protocol (SOAP) (2003), http://www.w3.org/TR/soap
24. Protégé. http://protege.stanford .edu
25. Doan, A.H., Madhavan, J., Dhamankar, R., Domingos, P., Halevy, A.: Learning to match ontologies on the Semantic Web. VLDB Journal, 303–319 (2003)
26. Hayes, J., Gutierrez, C.: Bipartite Graphs as Intermediate Model for RDF. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 47–61. Springer, Heidelberg (2004)
27. Hu, W., Jian, N.S., Qu, Y.Z., Wang, Y.B.: GMO: A Graph Matching for Ontologies. In: K-Cap 2005 Workshop on Integrating Ontologies 2005, pp. 43–50 (2005)
28. Tous, R., Delgado, J., Vector, A.: Space Model for Semantic Similarity Calculation and OWL Ontology Alignment. In: Bressan, S., Küng, J., Wagner, R. (eds.) DEXA 2006. LNCS, vol. 4080, pp. 307–316. Springer, Heidelberg (2006)

CSP Transactors for Asynchronous Transaction Level Modeling and IP Reuse

Lilian Janin and Doug Edwards

Advanced Processor Technologies GroupSchool of Computer Science,
The University of Manchester, UK
{lilian.janin,doug.edwards}@manchester.ac.uk

Abstract. In synchronous circuit design, new levels of abstraction above RTL allow the designer to model, simulate, debug and explore various architectures more efficiently than before. These are known as transaction level modeling. The translation between signals at different levels of abstraction is performed by pieces of code called transactors, mainly for the purpose of simulation. This paper identifies a set of asynchronous abstractions suitable for asynchronous transaction level modeling. Based on these models, we show that asynchronous CSP-based transactors can bring many more benefits than their synchronous counterparts, while being simpler to describe. We show how they can be used to automatically generate complex SystemC templates and hardware-software links, and automatically build network-on-chip interfaces facilitating IP reuse in embedded systems. Tools were developed after the techniques described in this paper. They are used in a case study to describe an asynchronous IP from transaction levels to RTL, demonstrating the automatic generation of various complex parts of the design and the minimum amount of specifications required from the designer.

1 Introduction

Chip design is becoming increasingly complex with billions of transistors and dozens of IPs being now commonly placed on single chips. In order to tackle this complexity, Electronic System Level (ESL) design has been developed extensively for the last five years, aiming at exploiting IP reuse and integrating higher levels of abstraction than traditional HDLs (Hardware Description Languages). Designing at these new levels of abstraction above RTL is known as transaction level modeling (TLM).

Over the same period, GALS (Globally Asynchronous, Locally Synchronous) techniques have made their way into chips as they provide an easy way to interconnect multiple modules with distinct clock domains, simplifying many synchronisation problems [1]. Implementing a GALS system using ESL tools is, in theory, simply a matter of linking 'locally synchronous' IPs to the 'globally asynchronous' interconnect IP (asynchronous buses or asynchronous networks on chip (NoC) such as CHAIN [2]). Although the interconnect IP is, for a GALS system, internally asynchronous, its interface is fully synchronous (devoid of asynchronous channels) in order to be connected directly to the ports of synchronous IPs. This allows ESL tools to handle only synchronous elements. Indeed, actual ESL tools are all designed to facilitate synchronous connections.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 154 – 168, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Nonetheless, it could often be beneficial to use IPs with asynchronous interfaces. An SRAM, for example, is an internally asynchronous device which sees its performance degraded when connected to the synchronous ports of an asynchronous NoC via a synchronous wrapper. This performance degradation could easily be avoided by using asynchronous interfaces to connect the SRAM IP block directly to the asynchronous NoC. One difficulty tackled in this paper is the description of these asynchronous TLM interfaces.

Another important aspect of ESL design tools and synchronous IPs is the integration of a hierarchy of models that are reusable across levels of abstraction. As specified in the OCP-IP [11] synchronous TLM proposal, tight relations from one level of abstraction to the next ensure that decisions made in the SystemC domain are carried through to the RTL domain faithfully, removing any chance for translational error and ensuring that simulations at various levels of abstraction are consistent with each other. These relations, although not inherently synchronous, are usually described using properties specific to synchronous interfaces. This paper identifies a set of properties that allow asynchronous models to be described consistently through abstraction levels. When simple properties cannot be versatile enough to describe all the possible configurations, more complex descriptions based on programming languages are introduced. This is the case with transactors, pieces of code performing the translation between signals at different levels of abstraction.

Section2 provides an overview of the abstractions proposed by the industry as standards for synchronous circuits. It introduces the transactors, and exposes the actual limitations of synchronous TLM and transactors and how asynchronous techniques are able to help. Section 3 defines a set of asynchronous abstractions suitable for asynchronous transaction level modeling. Based on these models, Section 4 shows how asynchronous transactors can be efficiently written and how they can be used to automatically generate SystemC templates, network-on-chip interconnect for IP reuse and hardware-software cosimulation links. Section 5 describes a practical example, where an asynchronous RAM IP is easily described from TLM to RTL using the tools developed after the techniques described in this paper.

2 Related Work

With the advent of ESL design, the various levels of design abstraction above RTL are referred to as Transaction Level Modeling. The primary goal of TLM is to dramatically increase simulation speeds, while offering enough accuracy for the design task at hand. The increase in speed is achieved by the TLM abstracting away the number of events and amount of information that have to be processed during simulation to the minimum required. For example, instead of driving the individual signals of a bus protocol, the goal is to exchange only what is really necessary: the data payload. TLM also reduces the amount of detail the designer must handle, therefore making modeling easier.

2.1 Transaction Level Modeling: Synchronous Abstractions

The two prevalent industry approaches are from OSCI (Open SystemC Initiative) and OCP-IP (Open Core Protocol -International Partnership): the former has focused its

TLM on the software development concerns, while the later has focused on SOC integration concerns. OSCI's TLM leads to a top-down methodology, where the design architect describes its design at a software level, via untimed software calls, and gradually refines it down to the RTL level. OCP-IP's TLM, on the other hand, leads to a methodology to specify reusable IPs through a bus-centric hierarchy of models seamlessly interconnecting via a common "socket" interface at different levels of abstraction. In spite of these different origins, both TLMs converge on several aspects. Both OSCI and OCP-IP define three levels of TLM above RTL, the former gradually refining an initial software-level model towards RTL and establishing a software-like top-down design methodology, and the latter gradually abstracting away the details of the communications between modules for IP reuse and easy on-chip interconnect. It is interesting that two initiatives coming from different fronts led to similar definitions.

2.2 Synchronous Transactors

One of the main objectives of ESL design is the transparent interoperability between IPs at different levels of abstraction. Transactors were created for this purpose: they are pieces of code performing the translation between transactions at different levels of abstraction. These transactors receive stimuli and translate the activity to the data structure, call-structure and temporal mechanisms used by design models at other abstractions. This enables each component in a system to communicate using the interface semantics native to its design abstraction. In a mixed abstraction level simulation environment, transactors are used to connect blocks modeled at different abstraction levels for simulation in different contexts of speed/accuracy. One main application of transactors is for testbenches: creating a transactor connected to a testbench makes the testbench insensitive to changes in the design. Even if the top-level interface of the design under verification is modified, only transactors have to be updated.

Traditionally, transactors were written as two unidirectional components or pieces of code: a *driver* from high to low levels and a *monitor* from low to high level. These allow test library functions written in an abstract, untimed way to exercise and check models at different abstraction levels. Transactors are often written in the same language as the one used to describe the models. When created manually, the process is long and error-prone, and is often as complex and time consuming as the creation of the RTL model itself. Moreover, transactors must be updated as the design interface changes, leading to many manual iterations. As transactors are complex to design, those created by hand often only connect two levels of abstraction. However many design simulations today have several abstractions integrated together. Consequently, handling only two abstraction levels with a transactor is not sufficient. Complex transactors that can handle interconnection of multiple levels of abstraction need to be automatically generated, and this requires the capture of the interface abstractions in a single formal description.

Recently, a few initiatives tried to define new languages for describing transactors more efficiently and for generating the simulated code automatically [3, 4]. A formal description of an interface abstraction hierarchy captures the temporal, data and behavioural aspects of an interface protocol at multiple levels of abstraction along with the mapping between the levels. Transactors can be generated from these descriptions to drive and monitor communications at various levels of abstraction.

Balarin and Passerone [5] proposed a methodology where interface protocols are specified formally in a way that is very similar to assertions used in verification, using regular expressions from the *Property Specification Language* [6]. Transactors are automatically generated from such a specification. Many transactors may be generated from a single interface specification, depending on which part of the design is being verified, what modes of the interface are being exercised, and which verification technology is being used (e.g. simulation vs. acceleration). In addition, formal interface specifications can be used as assertions and verified either statically or dynamically.

2.3 Proposed Improvements to the Synchronous TLM Models

This section presented the levels of abstraction defined as standards for synchronous transaction level modeling. These initiatives are extremely valuable as they standardise the development of synchronous IPs. However, they suffer a few drawbacks, which can be eased using asynchronous techniques.

OCP-IP's TLM defines a standard way to specify interconnectable -and thus highly reusable - IPs by defining an abstract bus interface which the designer needs to follow. The downside is that the interface of the IP is constrained by the TLM model. We claim that an asynchronous IP can rely on an automatic serialisation-deserialisation of any asynchronous interface, allowing IPs to *plug&play* to a network on chip without imposing any constraint to the interface.

Transactors provide a transparent translation between different levels of abstraction, enabling mixed-mode simulation, verification across levels, etc. Unfortunately, even using recent specification techniques to describe transactors more efficiently, their design often takes months [3]. The definition and common usage of standard asynchronous protocols makes asynchronous communications easier to describe. We show how asynchronous transactors can be developed very efficiently using existing languages.

OSCI have based their design methodology on SystemC. SystemC is appropriate for system-level design and transaction level modeling: designers can describe high level interfaces, structures and algorithms at the same time as delays and bit-level signalling. SystemC's main drawback is the complexity for the designer to describe the many required classes and to handle C++ subtleties such as exporting the abstract interfaces via inheritance in ports and channels, leading to a huge amount of work before being able to start describing the real circuit. We show how OSCI's methodology can be applied to asynchronous design and how the initial complexity of SystemC descriptions can be bypassed by using code generators based on the transactors.

The next section describes how we identified a set of abstractions applicable to asynchronous interfaces and how the translation from one level to another can be automated.

3 Asynchronous Transaction Level Modeling

The first task before being able to describe asynchronous IPs and transactors is to identify the levels of abstraction which will be useful for asynchronous transaction level modeling.

From the previous description of synchronous TLMs, we can see that synchronous communications, based on clock cycles and independent wires, are at the same time very constrained (by the clock signal) and free to implement any protocol (each wire is left to the imagination of its creator). This led to abstractions based on clock cycles: groups of events belonging to the same clock cycle and periodic events repeating every n cycles are bundled together. Asynchronous communications, on the other hand, are not constrained by a clock, and therefore abstractions cannot be based on it. However, a set of major asynchronous protocols exists, which are used in most designs, and from which a set of abstractions can be defined. This section focuses on the definition of abstraction levels based on these asynchronous protocols.

3.1 Asynchronous Abstractions

Asynchronous circuit design, based on asynchronous communications, has its roots as a high level model: the CSP (Communicating Sequential Processes) language [7], created by Hoare for describing patterns of interaction. Several CSP-based HDLs [8, 9] have been created over the years to describe and synthesise asynchronous circuits. While synchronous languages are striving to raise their level of abstraction while keeping enough control over the generated transistors, asynchronous designers have striven for years to synthesise low-level transistors from their high-level communication patterns. Asynchronous design might therefore be able to shine at the transaction level, by directly exploiting the CSP abstractions it originated from.

In synchronous TLMs, the Programmer View (PV) abstraction, based on software function calls with no communication events and no timing information, is the basis of transaction level modeling. As it is not related to any clock model, it can be defined and used in exactly the same way in an asynchronous TLM.

In OSCI's second abstraction, PV+T, timing information is added. Delays are a big problem in asynchronous simulation. In a synchronous design, every component is periodically resynchronised to the clock, minimising the imprecision by resetting any skew. In an asynchronous design, timing errors continuously accumulate, drifting to very imprecise values. Tackling the problem of asynchronous timing models is beyond the scope of this paper. So far we decided not to specify timing as part of any level of abstraction. Timing information can however still be associated to events at any level of abstraction, providing improved simulation accuracy.

In both OSCI's and OCP-IP's TLMs, the lowest abstraction above RTL is where communications are abstracted as "transfers" identical to asynchronous CSP communications. An asynchronous TLM can therefore be restricted to RTL, CSP and PV levels.

3.2 Abstraction Properties

In the context of transaction level modeling, and to enable communications to be automatically propagated and translated across levels of abstraction, we aimed to define levels of abstraction of asynchronous communications such that, given a transaction at the highest level of abstraction, a mixed-mode simulator is able to automatically generate a valid set of lowest level signals, and vice versa: given a set of low

level signals, an automated process exists to regroup them appropriately to reconstitute a coherent high level transaction. For this, we identified a set of properties able to define the transitions from one level of abstraction to the next.

Such a set of properties should allow, for instance, to convert a handshake transaction from an abstraction where data validity is not taken into account (the data becomes valid at exactly the same time as the handshake request is raised) to an abstraction where data validity is considered. Depending on the specified data validity scheme, the 'converter' would need to separate the reception of the data from the reception of the request signal, possibly delaying one of them to satisfy timing validity and inserting an 'invalid data' event at the appropriate time (Fig 2). This validity scheme would be a property of the abstraction level, which should be provided by the IP connected to such a channel when simulated at or through this level of abstraction.

Fig 1 shows the properties obtained by considering the difference between the PV level and the RTL level. It can be verified that setting these properties to false or zero makes for a very abstracted high-level description of the transaction, while setting these properties to real values makes for a low-level transaction. Each of these properties can be enabled or disabled, and the combined state of the properties defines a specific level of abstraction. Using these abstraction properties, Fig 3 shows how one transaction can be simulated and converted across the abstraction spectrum.

The properties can be grouped into 4 categories: Function, Structure, Protocol and Timing (Fig 1). Structure properties specify the interface of the IP blocks in terms of I/ Os, from abstract data type transfers to physical wires. Protocol properties define the succession of events being sent or received by the IP, therefore affecting its dynamic behaviour. Structure and Protocol properties are discrete properties (often just a boolean choice). They specify a precisely defined arrangement, usually observable in the final physical circuit. On the other hand, Timing properties are estimates of real world delays and can take any real value, while Function properties define the relationship between a structural entity (such as a wire) and a functionality. For example, a RnW wire would be activated to indicate a read function and deactivated to request a write. As mentioned previously, timing properties are not included in our current model. The three lower protocol properties happen to correspond exactly to the definition of handshake protocols. Transactions were the handshake protocol is abstracted away characterise the CSP abstraction, where data is sent over individual channels following a two-events protocol: request sent with data followed by acknowledge (for a push channel), and where the interleaving of different channels is specified.

4 Asynchronous Transactors

A transactor is a process able to automatically translate a signal or set of signals from one level of abstraction to another. We defined a TLM in the previous section with the aim to facilitate these translations for asynchronous IPs. A set of simple properties were proposed, that are able to direct the conversion between two contiguous levels of abstraction. Based on the levels of abstraction presented in the previous section, this section shows why asynchronous transactors are needed, how they can be written efficiently and how

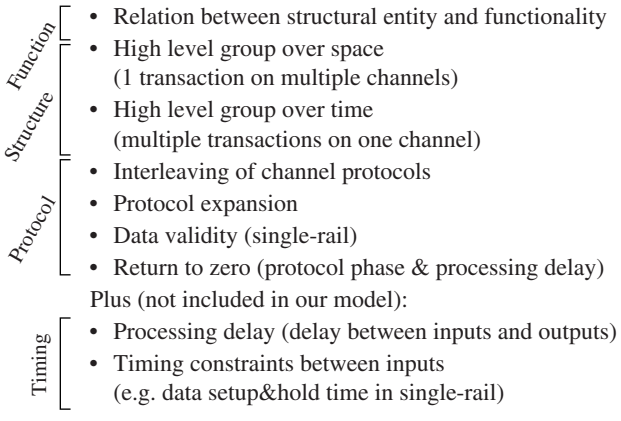


Fig. 1. Abstraction properties of a handshake transaction

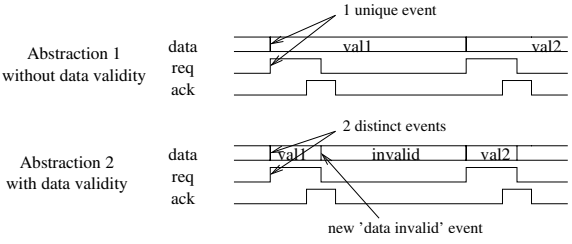


Fig. 2. Abstraction property example: Data validity

they can be used to automatically generate SystemC templates, network-on-chip interconnect for IP reuse and hardware-software cosimulation links. Our techniques are aimed at minimizing the amount of specifications required from the designer.

4.1 Automatic Translation Between Abstractions

Although we referred earlier to transactors as 'pieces of code', the aim is to perform translations across abstractions using the simplest possible specifications. For this reason, in order to automatically translate transactions between different levels of abstraction, each abstraction property is assigned a set of possible values. For example, data validity can be defined to be broad, early or late. These values are packaged with the IPs that require their signals to be translated at or through the associated level of abstraction. The major translations associated to our TLM are RTL to CSP, CSP to RTL, CSP to PV and PV to CSP. Each major translation is made of a succession of minor translations corresponding to each abstraction property.

The RTL to CSP translation is the simplest one: from a full set of events issued from the RTL simulation, we only need to extract the data payload, the initial request indicating the start of the CSP function call (send or receive) and the final acknowledge

indicating the return of the CSP call. This conversion is processed in three steps, each one of them disabling in turn one abstraction property defining the handshake protocol: first, if defined in the protocol in use, the Return To Zero phase is removed (all the events not part of the RTZ are passed to the next stage), then data validity is "removed" -in fact, it is set to a default state where the data becomes valid with the incoming request (for a push channel) and never becomes invalid. Finally, the rest of the protocol is removed, leaving just enough information to generate the CSP call.

CSP to RTL, the opposite translation, is a bit more complex as information is added during the conversion process: instead of being disabled, abstraction properties are enabled and assigned real values. At the CSP level, the initial request, data payload and final acknowledge are known. From the definition of the handshake protocol, a template specifies how to rebuild the data signals at the wire level. All missing signals are introduced between the initial request and the final acknowledge in three consecutive step (Fig 3). In future work, timing constraints will be used at this stage to insert appropriate delays between constrained signals. For the moment, delays are hard-coded.

CSP to PV and PV to CSP translations are more complex due to the need to map abstract functionalities to channels, and the need to specify the interleaving of channel events.They must be handled by transactors specified in a language featuring these characteristics.

4.2 CSP-Based Transactors

When invoked from the PV level, the task of a transactor is to specify the sequence, parallelism and interleaving of the different channels involved in a transaction. Although destined to other uses, a few freely available languages have been designed in the past to describe such dependencies: TAST-CHP [8], CH and Balsa [9]. They are all CSP-based, which is ideal for PV to CSP translations. Moreover, as they were designed for synthesis, the syntax and grammar they use to describe channel interleavings is precise down to the RTL level. Balsa was created for asynchronous design and has the advantages of being synthesisable and being a traditional, easy-to-learn, Verilog-like imperative language.

The interface of a PV-CSP transactor needs to combine the interfaces of the CSP and PV levels. PV level is represented by the signature of the function call corresponding to the transaction. CSP level is represented by the channels involved in the transaction. As a CSP-based language, Balsa is able to describe the CSP, but not the PV interface. Transactor interfaces are therefore specified by prefixing a C-style PV function interface to a Balsa block (as shown in the code example from Section 5 Step 3). In practice however, the Balsa CSP interface of a transactor can be generated from an earlier specification of the channels (Section 5 Step 2), eliminating the need to be repeated in the transactor.

The current description language for a transactor's body is only a subset of the Balsa language. There are two reasons for this: although the Balsa synthesiser can generate structural Verilog for any language construct, our generation of behavioural SystemC code is new and we have only implemented the constructs that were useful in our applications. The second reason concerns the invertibility of the language.

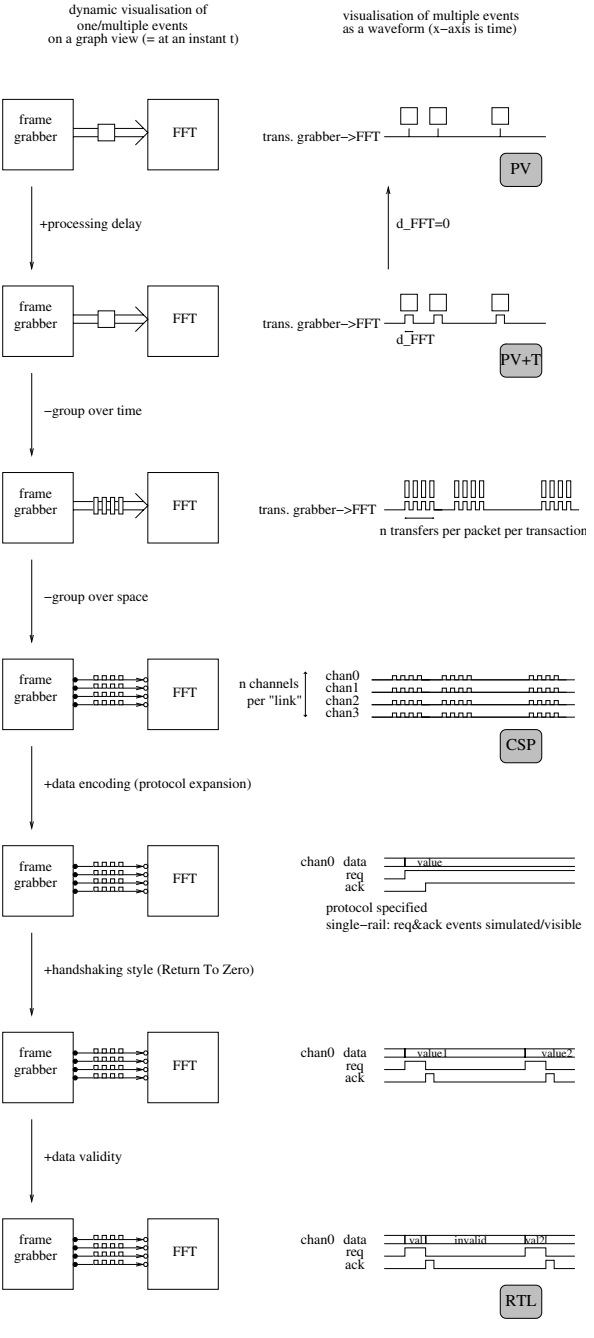


Fig. 3. From PV transaction to RTL signals

Most of the time, both PV to CSP and CSP to PV transactors are needed, and therefore two "reciprocal" Balsa descriptions need to be written (such as the two descriptions shown in Section 5 Step 3). In order to reduce the amount of work for the designer, one tool automatically generates a transactor's reciprocal description from its description.

CSP transactors enable a variety of automated code generation related to IP block interfaces. We describe three of them: Translator code generators, network on chip adapters and hardware-software links.

Translator code generators. In order to achieve the best simulation speed, the Balsa-described transactors are automatically converted to the same language as one of their attached components. If the chosen component is described in Balsa (component at CSP level), a direct Balsa simulation of component+transactor can be used. If the component uses Verilog (RTL level), a Verilog transactor is synthesised from the Balsa transactor using the Balsa synthesis tools. Finally, if the component uses SystemC (PV or CSP level), behavioural SystemC code is generated from the Balsa transactor. The main difficulty of this generator is also its main feature: fine-grained threads.

The Balsa language allows the easy description of very fine-grained threads, such as blocks containing a single assignment. On the other hand, threads in SystemC require a lot of "spaghetti coding", as each thread needs to be implemented in a separate class method. Automatically generating this kind of code from Balsa allows a clear and concise description of parallel code, which is often needed in asynchronous transactors.

Network on chip Plug&Play. One of the main motivations of System Level Design is IP reuse: describing an IP than can be integrated without any modification in other environments. OCP-IP defined a set of interfaces abstracting bus interfaces. IPs adopting these interfaces can then be connected easily to any bus. The downside with this method is that the designer is constrained to use these specific interfaces.

Asynchronous transactors are able to convert any CSP interface defined by the designer to a high-level transaction assimilable to a datas tructure. This means that transactors are able to convert CSP transactions to flows of bits. These flows can be transferred over any serial bus. In particular, the Balsa transactors can be used as adapters (serialisers and deserialisers) to automatically connect the IP to a network on chip.

Hardware-software link. The serialisation feature of the transactors can also be exploited for hardware-software links. The first application is to connect an emulator board to a software simulator via a serial link. On the hardware side (emulator board), transactors are synthesised in the same way as if they were connecting the IP to a hardware NoC. On the software side, transactors are simulated in the same way as if they were connected to a software NoC model. This configuration relies on a serial link connection between the simulator and the board.

The second application is for hardware-software (HW-SW) co-design. The automatic serialisation of the HW-SW interfaces would greatly facilitate the exploration of various HW-SW partitions.

4.3 Interface and Code Template Generators

After having described transactors, we realised that an important problem of asynchronous IP design could be solved: the complexity of writing SystemC code. Not only SystemC templates can be generated from the description of the interfaces at various levels of abstraction, but also transactors make it possible to automatically deduce an interface from its description at another level of abstraction. A step by step methodology (such as the one used in the case study presented in Section 5) shows that we can generate a complete SystemC template from interface descriptions at the PV and CSP levels, the RTL interface being deduced from CSP interface associated to the CSP to RTL abstraction properties, and large code templates being built from the transactor descriptions.

Multiple languages are used in a typical flow: SystemC at PV and CSP levels, Balsa atCSPlevelandVerilogatRTLlevel.Theycanbesynthesisedtolowerlevelsforsystem implementation on a chip or FPGA prototyping.

5 RAM IP Demonstrator

Traditionnally, synchronous transactors have been used to interface IPs with buses or NoCs using complex protocol and routing strategies. The asynchronous transactors presented in this paper are aimed more generically at linking any IP together. Some of these IPs may be busses or Nocs, but this is not compulsory, and pipelines of computational IPs can be built with asynchronous transactors. This section presents a simple example based on accesses to a RAM. This example, although lacking complexity, already implies a large amount of automatically generated code.

The full source code, interface descriptions and tools can be found in [10].

Step 1: high-level description of the RAM IP with test harness
From a description of the RAM interface at the PV level, we can generate a SystemC template that is easy to start a project with. A typical software-level RAM interface is:

```
module RAM { void write (int addr, int val); int read (int addr); }
```

Two classes, RAM and TestHarness, are automatically generated with a top-level sc_main function. This function instantiates the two components, connects them together and starts the SystemC simulation.

Before being able to simulate something useful, the behaviour of the RAM and test harness needs to be specified: the two method declarations (read and write) can be filled in and, in this particular case, a class variable for the memory array is declared:

```
constant MEMORY_SIZE 256;
int memory[MEMORY_SIZE];

virtual void write(int addr, int val) {
        cout << "RAM received write ("
                << addr << ", "
                << val << ")\n";
        memory[addr] = val;
}
```

```
virtual int read(int addr) {
        cout << "RAM received read ("
                << addr << ")\n";
        return memory[addr];
}
```

Writing the test harness is also easy, simply by filling up the run method of the test_harness class:

```
void run() {
      // use abstract interface to communicate with RAM
      for (int i=0; i<MEMORY_SIZE; i++)
            link_to_RAM->write (i, i+10);
      cout << "Processor read at address 2 => " << link_to_RAM->read(2) << eol;
}
```

A minimum amount of coding results in the simulation of our RAM IP. A good achievement for whoever tried to do this with SystemC before.

Step 2: Refinement of the interface to the CSP level

Here we enter the asynchronous domain. The data type of each channel needs to be described. The description format of the interface is now based on the Balsa language. The CSP implementation of the RAM is based on four channels: a 'command' channel, which indicates whether a write or a read command is issued, and the self-explicit channels 'address', 'value_write' and 'value_read':

```
enum command_type is CMD_WRITE, CMD_READ end
input command:      command_type;
input address:      int;
input value_write:  int;
output value_read:  int;
```

In the same way as at the PV level, SystemC code at the CSP level can be generated from this interface, for both the RAM IP and a test harness. The RAM class is now defined with SystemC sc_port ports publishing a CSP interface:

```
template <class T>
class CSP_send_if :
virtual public sc_interface {
 public:
  virtual void send(T) = 0;
  virtual bool probe() = 0;
};

template <class T>
class CSP_receive_if :
virtual public sc_interface {
 public:
  virtual void receive(T&) = 0;
  virtual bool probe() = 0;
};
```

```
class RAM : public sc_module {
 public:
      sc_port <CSP_receive_if
            <command_type> > command;
      sc_port <CSP_receive_if <int> > address;
      sc_port <CSP_receive_if <int> > value_write;
      sc_port <CSP_send_if <int> > value_read;

      // Constructor
      SC_CTOR(RAM) { SC_THREAD(run); }

      void run() { }
};
```

At the PV level, the simulation was based on a single thread (the test harness) and standard function calls. At the CSP level, each instantiated module is a thread, thus modeling hardware more closely. The two PV methods 'read' and 'write' are replaced by a single 'run' thread method. A typical implementation of the main method

waits for a command to be received, and completes a write operation if the command is CMD_WRITE or a read operation if it is CMD_READ: (this code is user-written here, but we will see later that it can be generated automatically).

```
void run() {
  for (;;) { // loop forever
    enum command_type cmd;
    int addr, val;

    command->receive(cmd);
    address->receive(addr);

    switch (cmd) {
    case CMD_WRITE:
      value_write->receive(val);
      cout << "RAM received write ("
        << address << ", " << val << ")\n";
      memory[addr] = val;
      break;

    case CMD_READ:
      cout << "RAM received read ("
        << addr << ")\n";
      value_read->send (memory[addr]);
      break;
} } }
```

The test harness is described at the CSP level in the same way as the RAM 'run' method. However, another possibility is to reuse our previously written test harness at the PV level and rely on a PV to CSP converter. This is the role of the transactor.

Step 3: Description of the transactor

The following transactor is able to perform the translation between PV's read and write calls and CSP channels' send and receive methods.

```
void write (int addr, int val):
{
    command <- CMD_WRITE ||
    address <- addr ||
    value_write <- val
}
```

```
int read (int addr):
{
    command <- CMD_READ ||
    address <- addr ||
    return <-value_read
}
```

This transactor is obviously able to translate PV read and write calls to CSP send and receive calls. This allows us to connect the PV test harness to the CSP RAM.

What is really interesting is that, from this PV to CSP transactor, it is possible to generate most of the CSP level implementation of the RAM automatically. In our example, the Balsa reciprocal transactor (CSP to PV) is generated as below:

```
select command, address then
    case command in
        CMD_WRITE:
            select value then
                extern write (address, value_write)
            end
        CMD_READ:
            value_read <- extern read (address)
    end
end
```

The SystemC code generated from this new transactor is exactly the 'run' method implemented by hand previously.

Step 4: Refinement/synthesis to RTL
In order to go from CSP to RTL level, the handshake protocol used by each channel needs to be specified. A possible description is as follow (using multiple protocols):

```
command:      push    dual-rail     2-phase
address:      push    1-of-4        4-phase
value_write:  push    single-rail   4-phase
value_read:   push    dual-rail     4-phase
```

From these specifications, an RTL-level SystemC interface can be generated. An RTL-level 'run' method can also be automatically generated. However, using SystemC at the RTL level involves some efforts in order to interleave the handshakes appropriately and the designer ends up with a non-synthesisable RTL description. This task can be done a lot more easily by using a language such as Balsa to synthesise the RTL automatically from the CSP-style description. Another useful alternative is to use a HDL such as Verilog for the description at the RTL level. This is the path explored for this application: we decided to target an FPGA and use its RAM structures to implement the RAM IP. This allows us, in the future, to access the FPGA's RAM structures from any language, even if this language's synthesis tools would not automatically use these on-board RAMs. This is for instance the case with the Balsa language.

The Verilog interface template is generated from the IP interface specifications. Even more helpful, a template of the Verilog implementation can be generated from the transactor description: this is a synthesised version of the Balsa "CSP to PV" transactor, preceded by protocol converters able to adapt each channel's RTL signals to the CSP input of the transactor. Although it does not result in the most efficient implementation, this generated template is ideal to start the RTL level design and can be optimised by hand if necessary. The template automatically takes care of sending requests on outgoing channels and waiting for incoming requests on incoming channels. The only part to be filled in is the handling of the read and write commands, but this time in Verilog at the RTL level.

6 Conclusions

We identified the PV and CSP levels as a set of asynchronous abstractions suitable for asynchronous transaction level modeling above RTL. Based on these models, we showed that asynchronous CSP-based transactors can bring many more benefits than their synchronous counterparts, while being simple to describe. We showed how they can be used to automatically generate complex SystemC templates and hardware-software links, and automatically build network-on-chip interfaces facilitating IP reuse. Both CSP to PV and PV to CSP translations were generated from a unique transactor description. In order to do this, we were able to make a useful subset of the Balsa language invertible. Tools were developed after the techniques described in this paper, and a case study demonstrated the automatic generation of various complex parts of the design and the minimum amount of specifications required from the designer.

References

[1] Moore, S., Taylor, G., Robinson, P., Mullins, R.: Point to Point GALS Interconnect. In: Eighth International Symposium on Asynchronus Circuits and Systems (ASYNC'02) (2002)
[2] Bainbridge, W.J., Furber, S.B.: CHAIN: A Delay Insensitive CHip Area INterconnect. IEEE Micro special issue on Design and Test of System on Chip 22(5), 16–23 (2002)
[3] Parker, D.: Transactor generation using the CY language. SpiraTech Limited
[4] TransactorWizard, http://www.sdvinc.com
[5] Balarin, F., Passerone, R.: Functional Verification Methodology Based on Formal Interface Specification and Transactor Generation. Design, Automation and Test in Europe (2006)
[6] Property Specification Language: Reference Manual http://www.accellera.org/pslv101.pdf
[7] Hoare, C.A.R.: Communicating sequential processes. Comms of the ACM 21(8) (1978)
[8] Renaudin, M., Rigaud, J.B., et al.: TAST CAD Tools." ASYNC'02 Tutorial (2002)
[9] Edwards, D., Bardsley, A.: Balsa: An Asynchronous Hardware Synthesis Language. The Computer Journal 45(1), 12–18
[10] Balsamics, http://www.cs.manchester.ac.uk/apt/projects/tools/balsamics/transactors
[11] Open Core Protocol International Partnership, http://www.ocpip.org

A Robust Real-Time Message Scheduling Scheme Capable of Handling Channel Errors in Wireless Local Area Networks*

Junghoon Lee[1], Mikyung Kang[2], Gyung-Leen Park[1], In-Hye Shin[1], Hanil Kim[1], and Sang-Wook Kim[3,**]

[1] Dept. of Computer Science and Statistics, Cheju National University
[2] Information Sciences Institute East, University of Southern California
[3] College of Information and Communications, Hanyang University
{jhlee, glpark, ihshin76, hikim}@cheju.ac.kr,
mkkang@east.isi.edu, wook@hanyang.ac.kr

Abstract. This paper proposes and analyzes the performance of a robust real-time message scheduling scheme capable of handling channel errors by exploiting an auxiliary schedule on the IEEE 802.11 WLAN. While the primary schedule is derived from the legacy EDF (Earliest Deadline First) scheme, the auxiliary schedule is generated based on the slack and arrival times of each message stream. During runtime, with the support of a channel probing functionality, the access point dynamically switches the two polling schedules according to the channel status, giving a temporaily unreachable stream more chances to meet its time constraints. Simulation results show that the proposed scheme improves the success ratio by up to 8.56 % and performs well even if the length of error duration gets larger, demonstrating that it is also robust to the bursty errors.

1 Introduction

Wireless communication technology is gaining a wide-spread acceptance for distributed systems and applications in recent years[1]. As both speed and capacity of wireless media such as IEEE 802.11 WLAN (Wireless Local Area Network) increase, so does the demand for supporting real-time applications such as broadband VOD (Video On Demand) and remote monitoring system. Real-time messages are inherently time sensitive and their time constraints are given from their own applications. For example, in the wireless sensor network, the periodically sampled data are delivered to the appropriate station within a reasonable deadline to produce timely and meaningful response[2]. The message of sensor data has a hard real-time constraint that it should be transmitted within a bounded delay as long as the corresponding channel stays in an error-free state. Otherwise, the data are considered to be lost.

* This research was supported by the MIC, Korea, under the ITRC support program supervised by the IITA (IITA-2006-C1090-0603-0040).

** Corresponding author.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 169–178, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Not a few scheduling schemes for the real-time message have been proposed in various respects of its time constraints[3]. In soft real-time applications, a proportional share of network resources is prospective, as can be shown in fair queuing approaches[4]. On the other hand, a hard real-time message stream needs the guarantee from the underlying network that its time constraints are always met in advance of the system operation or connection setup, while the guarantee is provided based on a rigid priority driven scheduler. EDF (Earliest Deadline First), which dynamically picks a message that has the closest deadline among the pending ones, is known to be the most common scheduling policy for real-time message streams[5]. WLAN can also exploit EDF for message scheduling in various ways on the assumption that the communication pattern of sensory data is fixed and known *in priori*. For example, AP (Access Point) polls each station according to the offline EDF schedule built prior to the system operation.

As was pointed out for the fair packet scheduling case, however, it is not clear how well this algorithms will work for the wireless networks, since wireless channels are subject to unpredictable location-dependent and bursty errors, which make a real-time application fail to send or receive some of its real-time packets[6]. As a result, the reservation-based guarantee scheme alone is not sufficient. A message may be unable to use the reserved network time due to channel status fluctuation, no matter how sophisticatedly its reservation was made. Therefore, the real-time performance such as network throughput or deadline meet ratio depends on how to reallocate the unused or invalidated time slots as well as how to compensate hosts whose packets are corrupted or postponed behind its reservation. This paper addresses these problems based on the additional auxiliary polling schedule which can switch the primary one when a specific channel falls in a bad state.

The rest of this paper is organized as follows: Section 2 introduces related works on real-time message scheduling and then Section 3 explains basic assumptions on network and error models. Section 4 presents the proposed scheduling scheme in detail. After exhibiting performance measurement results in Section 5, Section 6 concludes this paper with a brief description of future works.

2 Related Works

We will begin with a description of IEEE 802.11 WLANs, as plenty of related researches on the wireless sensor network employ WLAN as their underlying network. Thanks to the inherent nature of a wireless medium, messages are broadcasted to all the nodes inside the cell[7]. Correspondingly, the IEEE 802.11 was developed as a MAC (Medium Access Control) standard for WLANs[8]. The standard includes a basic DCF (Distributed Coordination Function) and an optional PCF (Point Coordination Function) to support efficient delivery of both real-time and non-real-time messages. The DCF exploits CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol as the basic channel access mechanism to transmit asynchronous messages. Due to the contention-based nature, packet collisions are intrinsic to CSMA, and their resolving procedure

makes the network access unpredictable. Meanwhile, real-time guarantee can be provided only if a deterministic contention-free schedule is developed[9]. As a polling-based protocol without contention, the PCF enables the real-time station to deterministically access the common channel medium.

Real-time protocols on WLAN are typically based on a frame-structured access which consists of a contention part and a reservation part, whether they are implemented as a centralized or distributed method. As an example, Choi and Shin suggested a unified protocol for real-time and non-real-time communications in wireless networks[10]. A BS (Base Station) polls a real-time mobile station according to the non-preemptable EDF policy. Traditionally, an EDF algorithm assigns priorities to individual jobs belonging to a task with respect to their absolute deadlines. The status of a packet on the real-time mobile station can be estimated from the traffic characteristics submitted to the BS at the connection establishment time. The BS also polls the non-real-time message according to the modified round-robin scheme rather than a standard CSMA/CA protocol to eliminate message collisions. They also adopted the channel state prediction, transmission deferment, and retransmission based on the method of probing before polling[11].

As the most prominent example of a distributed dynamic priority scheduling mechanism, M. Caccamo et al. have proposed a MAC that supports deterministic real-time scheduling via the implementation of TDMA (Time Division Multiple Access) to apply the EDF scheme to WLAN[9]. Referred as *implicit contention*, their scheme makes every station concurrently run the common real-time scheduling algorithm to determine which message can access the medium. Each message implicitly contends for the medium through the scheduling algorithm, for example, with priorities, rather than explicitly on the physical medium, thus it can save power and time. However, for this implicit contention, every node must schedule all the messages in the network, making it difficult to scale to large networks.

As a non-EDF scheduling scheme, Carley et al. have proposed an interesting TDMA schedule scheme for wireless sensor networks implemented with a contention-free message scheduler[1]. They showed that a message set whose periods are harmonic always has a contention-free phase assignment so long as its utilization is less than or equal to 1.0. In addition, a message set can be made harmonic by reducing message periods by at most half. Their scheme enables the network to contain hundreds of nodes exchanging thousands of messages. However, they didn't take network errors into account at all.

3 Basic Assumptions

3.1 Network and Error Models

In WLAN, each cell, namely, BSS (Basic Service Set) is assumed to consist of an AP and multiple sensor stations as shown in Fig. 1. AP may be linked to a wired backbone or other wireless channel. In the latter case, AP has two transceivers that operate on different frequency levels, one for intracell communication and the other for

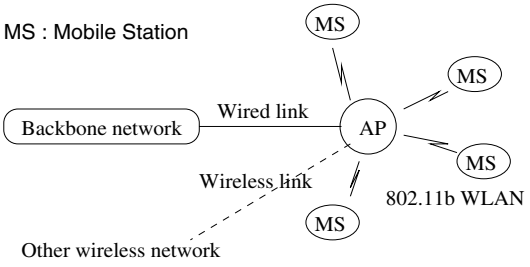


Fig. 1. Network model

intercell communication. Every station shares medium on the common frequency band and accesses according to the predefined MAC protocol. Each flow is ether an uplink (MS to AP) or downlink (AP to MS), while AP coordinates the overall network operations. This paper exploits the TDMA style access policy as in [1][6][10], now that the real-time guarantee cannot be provided without developing a deterministic access schedule, as well as the contention resolution via packet collisions consumes the precious communication energy.

Each station is associated with a channel which has either of two states, namely, *error state* and *error-free state* at any time instant. A channel is defined between each mobile and the AP, and it can be modeled as a Gilbert channel[12]. We can denote the transition probability from state *good* to state *bad* by p and the probability from state *bad* to state *good* by q, as shown in Fig. 2. The pair of p and q representing a range of channel conditions has been obtained by using the trace-based channel estimation. The average error probability and the average length of a burst of errors are derived as $\frac{p}{p+q}$ and $\frac{1}{q}$, respectively. A packet is received correctly if the channel remains in state *good* for the whole duration of packet transmission. Otherwise, it is received in error. Channels between the AP and respective stations are independent of one another in their error characteristics.

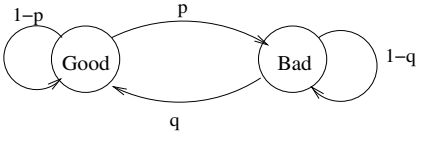


Fig. 2. Error model

3.2 Message Model

A station, currently inactive, can be activated by an upper layer query command that wants to monitor or process the data flow from the sensor station[13]. The query may also specify a sampling period as well as a precision level of needed data (hence, message length) on a station. In case of a change in the active flow set, the network schedule should be recalculated[5]. The destination of a message

can be either within a cell or outside a cell, while the outbound messages are first sent to the AP and then forwarded to the remote destination. Internal messages are also relayed by the AP. Hence, the AP also acts as another source of message stream with an exception that it is just virtually polled.

The traffic of sensory data is typically *isochronous* (or synchronous), consisting of message streams that are generated by their sources on a continuing basis and delivered to their respective destinations also on a continuing basis[5]. This paper follows the general real-time message model which has n streams, namely, S_1, S_2, ..., S_n, each of which generates a message of less than C_i at each beginning of its period, P_i. In addition, the first message of each stream arrives at time 0. Each packet must be delivered to its destination within D_i time unit from its generation or arrival at the source, otherwise, the packet is considered to be lost. Generally, D_i coincides with P_i to ensure that the transmission completes before the generation of the next message. However, sometimes, D_i is larger than P_i, and in this case, more errors can be recovered within the message deadline. Finally, the stream belongs to a specific station, so if a slot is assigned to a stream, it means that the AP should poll that station at its slot time.

4 Network Operation

4.1 Bandwidth Management

The network time is divided into a series of fixed-size slots, each of which is exclusively assigned to a real-time station, completely removing the possibility of packet collision, for the purpose of both real-time guarantee and power saving, as mentioned earlier. The slot length, say L, is as large as the basic unit of wireless data transmission and every traffic is also segmented to fit this slot size[9]. On WLAN, this TDMA-style network access can be implemented by making AP poll each station according to the predefined schedule, as well as making a polled node send for as large as L. The allocation scheme assigns each slot to a specific real-time stream, taking into account their time constraints. This assignment implies that a preemption occurs only at the slot boundary. Meanwhile, as every message is transmitted by the way of AP, be it intercell or intracell message, the AP knows whether a transmission has succeeded or failed. So the error recovery function is performed also under the control of the AP.

For the access point to decide the polling order for a stream set, each stream, S_i, submits a tuple of (P_i, C_i) information. At first, the schedulability of message streams is tested by the following sufficient condition[9]:

$$\forall_i,\ 1 \leq i \leq n\ \sum_{k=1}^{i} \frac{C_k}{P_k} + \Delta \leq 1.0,$$

which assumes that all the messages are sorted in increasing order of relative deadlines and that there are n streams. Also, Δ denotes the overhead term originated from the network management such as polling/probing overhead, beacon packet broadcast, interframe space, and so on. As Δ is fixed for a stream set, we can calculate per-slot overhead and merge it into C_i. If we let Δ_p the per-slot

overhead, the modified transmission time, C_i^{t+1}, can be calculated iteratively with the following equation until C_i^{t+1} becomes equal to C_i^t.

$$C_i^{t+1} = C_i^t + \lceil \frac{C_i^t}{L} \rceil \Delta_p$$

As a result, if we think Δ is absorbed into C_i, Δ can be assumed to be 0, enabling us to concentrate on the problem of slot allocation.

After acceptance test for a given stream set, the transmission is reserved for each stream. That is, the AP creates a polling table for the entire polling sequence[14]. The invocation behavior of a periodic task set repeats itself once every T time units, where T, called the *planning cycle* of the task set, is the least common multiple of the periods of all periodic tasks. Hence, we only need to consider all the task invocation in T. For S_i, every message arrival is estimated at 0, P_i, $2P_i$, and so on, while each invocation needs $\lceil \frac{C_i}{L} \rceil$ slots. With this estimation, the AP selects the message whose deadline is closest at each slot boundary and assigns the slot to the corresponding stream. If no slot request is pending at a slot boundary, the scheduler leaves it unassigned. Fig. 4 also shows the allocation example for the stream set, τ :{A(6,1), B(3,1), C(4,2)} whose utilization is 1.0.

Though this polling schedule can meet the time constraints of a given stream set, the AP may be unable to poll the stream at the scheduled time because the channel between itself and the stream has fallen to a bad state. In this case, the existing poll is invalidated. Without some compensation procedure, the invalidated stream will fail to meet its deadline. To overcome this situation, this paper proposes an auxiliary schedule shown in Fig. 3. The auxiliary schedule helps the runtime scheduler to pick alternative stream when it cannot poll the stream in the primary schedule. To build an auxiliary schedule, AP generates both arrival time and slack time of each slot in the primary schedule based on EDF policy. By E_t, we denote the stream AP will poll at time t. If E_t is M, A_t and L_t denote the arrival time and the slack time of M, respectively. Slack means the redundant time to the completion of a given task or message. It can be calculated by subtracting the current time and the execution time from the deadline. Let's assume that slot t is assigned to stream M, that is, $E_t = M$. In addition, M is unreachable at time t during the actual network runtime. If the poll to M is switched by a poll to another stream at $t^{'}$, and $t^{'}$ is less than or equal to the deadline of M, M has one more chance to be delivered within its deadline.

Based on this concept, we fill the auxiliary polling list as follows: for M at time t, from the stream located at $t + L_t$ down to $t + 1$, the scheduler checks whether the occupied message, say $M^{'}$, is already pending. If so, that is, the arrival time of $M^{'}$ is less than or equal to time t, $M^{'}$ is selected for E_t. $t + L_t$ is the farthest slot to which M can be moved. Fig. 4 shows the example. For time 2, the primary schedule is to poll stream C, while A_2 and L_2 are 0 and 1, respectively. The first candidate for the auxiliary schedule for slot 2 is the stream located in the primary schedule at time $2 + L_2$, namely, stream A. In addition, A_3 is 0 and is less than current time, 2, so we can conclude that A is already pending at time 2. After all, A is selected for the alternative schedule for E_2, namely, C.

```
for all t in the planning cycle
    t' = t + L_t
    while (t' > t)
        if (A_t' ≤ t)
            select E_t' as X_t
        t' ← t' − 1
    end while
end for
```

Fig. 3. The construction of an auxiliary schedule

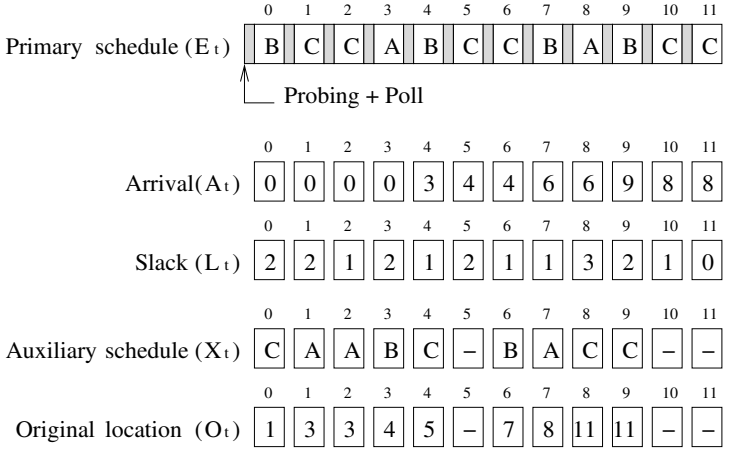


Fig. 4. Slot allocation and probing

4.2 Runtime Operation

At runtime, the AP has the primary and auxiliary schedules, and at each beginning of a planning cycle, the primary scheduling table is reloaded. In addition to E_t, A_t, and L_t, we further define X_t as the auxiliary schedule at time t, while O_t denotes the original location of X_t. As specified in the primary reservation table, each station is polled one by one, while the unallocated slot can be used for retransmission. The AP also maintains a retransmission queue ordered by the deadline of failed messages and a message entry will be automatically removed when its deadline expires. In addition, it is advantageous to probe a channel before polling or transmitting, as it can not only avoid a useless transmission over a bad channel, but also decrease the need for retransmitting packets. The channel estimation is achieved by transmitting a probing control packet to the scheduled station, which then returns the control packet, just like the RTS/CTS (Request To Send/Clear To Send) interaction[10]. If the AP does not receive the probing control packet correctly from the station, the channel is estimated to

be bad. However, even though the probing indicates the channel stays good, the ensuing transmission can fail if a state goes to bad during the transmission.

If the probing fails for a stream, the scheduler should select another one and try to poll, but at this time, without a probing, as the unlimited number of probe may invalidate the schedule which is based on the fixed size slots. If the transmission succeeds, the scheduler modifies some schedules. Fig. 5 summarizes the runtime operation steps.

```
t = 0
while system operation
    get a stream, E_t, from the primary polling table
    if E_t is unassigned
        get the stream from the retransmission queue and poll
        return
    probe the channel status of E_t
    if good
        poll E_t
    else
        poll X_t
        if successful
            place E_t in the slot time O_t
        else
            put E_t into the retransmission queue
    t ← (t+1) mod T
end while
```

Fig. 5. Description of runtime operation on AP

However, the proposed scheme has some margin to further improve the performance. Let's assume that C is in bad state from time 1 to 2 in Fig. 4. X_1 and X_2 are both A. In this case, as A may be already transmitted at time 1 in place of C, switch to X_2 at time 2 is meaningless. This situation requires the more sophisticated selection of the auxiliary schedule.

5 Performance Evaluation

This section measures the performance of the proposed scheme in terms of the success ratio according to the packet error rate as well as the average length of error duration, $\frac{1}{q}$ in Fig. 2, via simulation using the ns-2 event simulator[15]. The success ratio means the ratio of the number of successfully transmitted packets within their deadlines to the total number of generated packets. For the target stream sets, we fixed the length of a planning cycle to 24 as well as the number of streams to 3, and generated every possible stream sets whose utilization ranges from 0.2 to 1.0. Such sets have various utilization, slack distribution, and so on, as the number 24

has many integer divisors. We measured how many errors can be recovered by the auxiliary polling schedule, for the packet error rate of 0.0 through 0.5. Actually the packet error rate is a function of the average packet length and bit error rate, and we chose the packet error rate considering the error-prone wireless channel characteristics. Fig. 6 plots the measured result and compares the success ratio with the legacy EDF schedule. As shown in this figure, the performance gap gets larger with a higher packet error rate, by up to 8.56 %.

Fig. 7 plots the success ratio according to the average length of error duration. This parameter can influence the success ratio because the error can be recovered only if the slot is switched with one that gets out of the error duration. Hence, we measured the success ratio of the proposed scheme by changing error duration length from 0.0 to 4.0 slots. We further assume that each error duration exponentially distributes with those averages. In Fig. 7, the curve for the EDF remains constant, as the length of error duration does not affect the overall packet error rate. In contrast, as the length of error duration increases, the success ratio of the proposed scheme diminishes as expected. However, the curve shows that even in the large value of error duration, the proposed scheme can reasonably enhance the real-time performance of wireless sensor networks.

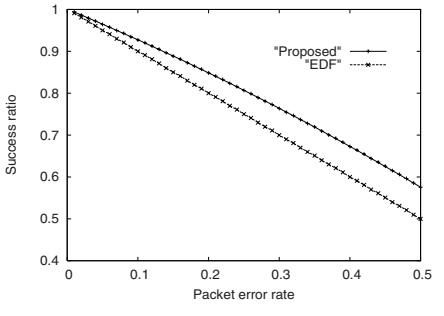


Fig. 6. Effect of error rate

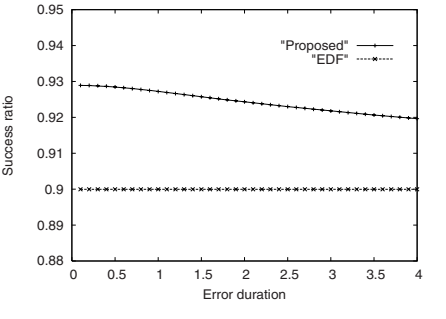


Fig. 7. Effect of error duration (slots)

6 Conclusions

This paper has proposed and analyzed the performance of a robust real-time message scheduling scheme capable of handling channel errors via an auxiliary polling schedule on wireless sensor networks built on top of IEEE 802.11 WLAN. Based on the EDF policy, the AP basically determines the polling schedule for the active stream set. Besides this primary polling schedule, the AP also creates the auxiliary schedule to cope with a situation when a scheduled stream is not reachable. To this end, the primary schedule also generates the arrival time and the slack time of a message slot. For each stream, the farthest one from the stream on the primary schedule is selected for the auxiliary polling candidate as long as it has already arrived. This allocation can cope with the relatively long error duration. With two polling schedules, the AP polls each station one by

one as specified in the primary reservation table, while the unallocated slot can be used for retransmission. If probing fails, the stream in the auxiliary schedule is selected. If this alternative transmission succeeds, the primary schedule is modified to give the invalidated stream one more chance to meet its deadline. The performance measurement result shows that the proposed scheme can improve the success ratio by up to 8.56 % while demonstrating that even in the large value of error duration, it can reasonably enhance the real-time performance.

References

1. Carley, T., Ba, M., Barua, R., Stewart, D.: Contention-free periodic message scheduler medium access control in wireless sensor/actuator networks. In: Proc. IEEE Real-Time Systems Symposium, pp. 298–307 (2003)
2. Li, H., Shenoy, P., Ramamritham, K.: Scheduling communication in real-time sensor applications. In: Proc.10th IEEE Real-time Embedded Technology and Applications Symposium, pp. 2065–2080 (2004)
3. Lee, J., Kang, M., Park, G.: Design of a hard real-time guarantee scheme for dual ad hoc mode IEEE 802. In: Syrotiuk, V.R., Chávez, E. (eds.) ADHOC-NOW 2005. LNCS, vol. 3738, pp. 141–152. Springer, Heidelberg (2005)
4. Vaida, N., Dugar, A., Gupta, S., Bahl, P.: Distributed fair scheduling in a wireless LAN. IEEE Transactions on Mobile Computers 4(6), 616–629 (2005)
5. Liu, J.: Real-Time Systems. Prentice Hall, Englewood Cliffs (2000)
6. Adamou, M., Khanna, S., Lee, I., Shin, I., Zhou, S.: Fair real-time traffic scheduling over a wireless LAN. In: Proc. IEEE Real-Time Systems Symposium, pp. 279–288 (2001)
7. Rangnekar, A., Wang, C., Sivalingam, K., Li, B.: Multiple access protocols and scheduling algorithms of multiple channel wireless networks. Handbook of Algorithms for Mobile and Wireless Networking and Computing (2004)
8. IEEE 802.11-1999: Part 11-Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (1999)
9. Caccamo, M., Zhang, L., Sha, L., Buttazzo, G.: An implicit prioritized access protocol for wireless sensor networks. In: Proc. IEEE Real-Time Systems Symposium (2002)
10. Choi, S., Shin, K.: A unified wireless LAN architecture for real-time and non-real-time communication services. IEEE/ACM Trans. on Networking 8, 44–59 (2000)
11. Damodaran, S., Sivalingam, K.: Scheduling algorithms for multiple channel wireless local area networks. Computer Communications (2002)
12. Shah, S.H., Chen, K., Nahrstedt, K.: Dynamic bandwidth management for single-hop ad hoc wireless networks. ACM/Kluwer Mobile Networks and Applications (MONET) Journal 10, 199–217 (2005)
13. Madden, S., Franklin, M., Hellerstein, J., Hong, W.: The design of an acquisitional query processor for sensor networks. In: ACM SINGMOD (2003)
14. Lee, J., Kang, M., Park, G., Kim, I., Kim, C., Park, J., Hong, J.: A real-time message scheduling scheme based on optimal earliest deadline first policy for dual channel wireless networks. In: Sha, E., Han, S.-K., Xu, C.-Z., Kim, M.H., Yang, L.T., Xiao, B. (eds.) EUC 2006. LNCS, vol. 4096, pp. 264–273. Springer, Heidelberg (2006)
15. Fall, K., Varadhan, K.: Ns notes and documentation. Technical Report. VINT project. UC-Berkeley and LBNL (1997)

Design and Implementation of a Tour Planning System for Telematics Users

Junghoon Lee[1], Euiyoung Kang[2], and Gyung-Leen Park[1,⋆]

[1] Dept. of Computer Science and Statistics, Cheju National University,
[2] Dept. of Computer Education, Cheju National University,
66 Jeju-daehakno, Jeju-City, Jeju-Do, 690-756 Rep. of Korea
jhlee@cheju.ac.kr, euiyoung1@hanmail.net, glpark@cheju.ac.kr

Abstract. Aiming at providing an efficient tour schedule to tourists driving with a telematics device, this paper designs and implements an intelligent tour planning system based on the personalized tour recommender that may generate lots of destinations. To overcome the problem of long response time due to the computation of $O(2^n \cdot n!)$ complexity solver, we used initial set reduction, distributed computing via MPI-based Linux cluster, and finally Lin-Kernighan heuristic. An user interface was also implemented on a portable device using the utility of embedded operating system. Performance measurement results exhibit that the tour schedule can not only be offered to the user within 5 seconds when the number of TPOIs is less than 22, but also find a schedule whose satisfaction degree is very close to the optimal value.

1 Introduction

[1]In September 2004, *Jeju Telematics City* enterprise has been launched in Korea, aiming at not just testing telematics devices and services but also accelerating their instantaneous deployment[1]. With this enterprise, many of rent-a-cars in Jeju province have been equipped with in-vehicle telematics, opening a way to provide such services as tour guide, navigation, safety service, entertainment, and so on. From the viewpoint of an embedded system, the telematics is a computing device within a car, necessarily having a radio interface and a GPS (Global Positioning System) receiver. While the GPS receiver makes it possible to offer various location-based services to the driver, service contents can be delivered to end-users via the radio interface, CDMA (Code Division Multiple Access) in Korea. As Jeju is one of the most famous tourist places in East Asia, the telematics system indispensably puts importance on the efficient method of managing and providing tour information, while one of the most promising services is the personalized tour planning system[2].

The Jeju province has many kinds of unique tourist attractions, or TPOIs (Tourist Point of Interests) from now on, including beaches, volcanoes, cliffs,

⋆ Corresponding author.

[1] This research was supported by the MIC, Rep. of Korea, under the ITRC support program supervised by the IITA (IITA-2006-C1090-0603-0040).

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 179–189, 2007.
© Springer-Verlag Berlin Heidelberg 2007

mountains, subsidiary islands and so on, while hosting many kinds of tour activities including ocean sport, golf, hiking, and the like within a relatively small area. As so many diverse tour plans are possible, the tourist needs the assistance from an intelligent tour planning system. Tourists expect that the system presents the useful information with a minimum user input. Particularly, in terms of telematics we are targeting at, it is necessary to minimize the user-system interaction, since the device has limited user interface capability and users possibly access it while they are driving. The personalization feature can meet this requirement, as it can minimize the user interaction with the telematics. As a result, per-person preference input mechanism is more appropriate than the per-TPOI scheme most other recommender systems employ[3].

In the telematics service scenario, an end-user submits necessary data through the telematics device, whether the service is executed within the telematics device or by a backend server which can process even more sophisticated decision logic, handling large volume of data. In case of a recommender system, a user inputs his preference data to the telematics before he starts his trip, then the device transfers these data to the remote server. The server generates and returns a tour plan to the telematics, which stores and displays the route according to the current location of the vehicle. Not to mention, the user may dynamically change his schedule due to the traffic condition change, prolonged stay time at some tour points. To cope with this situation, it is desirable to keep the user data in the server. It will reduce the amount of data exchange via CDMA network, also enabling background preparation in the high-performance server.

Generally, the tour recommender system consists of two steps: First, the system selects the candidate TPOIs sorted by their ranks or scores determined by the system-specific criteria and algorithms. The second step builds the subset from candidate TPOIs to generate a corresponding tour schedule, considering calculated ranks, given time constraint, user location, and current traffic condition. Computing an optimal route for multiple TPOIs is a superclass of TSP (Traveling Salesman Problem) which is known to be NP-hard, the number of candidate TPOIs being the most critical factor to the execution time[4]. However, Jeju province has many lookout points that can make a tourist satisfied without spending much time, so the number of candidates for a tour schedule can get increased. Additionally, when just the personal preference is given, multiple TPOIs may be tied at the same rank. It is necessary to add all tie-ranked TPOIs to the candidate list.

Meanwhile, it is not uncommon to improve the computation speed by means of a parallel or distributed computing environment such as MPI-based Linux cluster. This architecture has many additional advantages such as reliability, scalability, and so on. Particularly, it fits for the location-based service that repeatedly processes a large amount of data. In addition, efficient heuristics for TSP have been already developed and easily applicable. Based on such assertion, this paper is to design and implement an efficient tour planning system capable of maximizing the degree of user satisfaction. The proposed system consists of a master and slaves of the cluster, each one having TSP solver installed,

cooperatively yields a schedule that can be reachable within the given total tour length, for the TPOIs.

This paper is organized as follows: After issuing the problem in Section 1, Section 2 reviews related works. The data classification model is specified for the TPOI and user information in Section 3. Section 4 describes the TPOI recommendation method and tour planning system in detail. Section 5 demonstrates the performance measurement results of our system. Finally, in Section 6, the study is summarized with a brief description of future works.

2 Related Works

The recommender system is defined as follows[2]: Recommender systems are an attempt to mathematically model and technically reproduce the process of recommendations in the real world. First of all, this system filters off the unnecessary or less important information[5]. While there are various filtering methods, they can be classified into collaborative, content-based, and knowledge-based filtering schemes[6]. In the knowledge-based model, used most commonly, knowledge is expressed in the form of a detailed user model, a selection & suggestion model, and a rich description of the items to be suggested. Besides, there are also some interesting techniques such as data-mining, entropy, artificial intelligence, and agent schemes[7,8], while it is even possible to combine some of them to increase the accuracy of recommendation.

Maruyama et. al have proposed P-Tour system which generates a tour schedule according to the user input on (1) starting/returning locations and departure/arrival time of the tour, (2) candidate destinations of the tour, and (3) a relative importance degree of each destination[4]. They implemented a route search engine to obtain a semi-optimal solution using genetic algorithms. Afterward, this scheme was evolved to allow users to optimize their tour schedules under multiple conflicting criteria. However, in this approach, users do not only have to specify per-TPOI preference, but also the current traffic information is not considered, making it difficult to be used on the telematics system.

A TPOI recommender system has been proposed to provide personalized tourist information[9]. To provide a personalized recommendation, this system first classifies the related data into user information and TPOI information, then defines detailed attributes specific to each information category. Based on this data model, similarity between users and their preference are analyzed to decide a final recommendation. Though our paper is based on this system model, their system generates just the top-N recommendation result without taking into account the tour route or time constraint. In addition, the service lacks the capability of recommending via portable device such as a telematics or dynamically changing the schedule in the mid of tour.

The tour plan essentially starts from and ends at the customer's hotel as long as the customer does not change his hotel, so it can be considered to be a TSP Problem[1]. TSP is one of the widely studied NP-hard combinational optimization problems and can be described as follows: To begin with, let $G = (V, E)$

be a graph where V is a set of nodes and E is a set of links, and $C = (c_{ij})$ be the distance or cost matrix associated with E. TSP is to find the cheapest way of visiting all of the nodes and return to the starting point. Among plenty of heuristics to solve TSP, the most famous one is Lin-Kernighan's[10,11]. It is known for its efficiency in finding near-optimal results, while the core of this scheme consists of link exchange in a tour. A research web site provides diverse TSP solutions including Lin-Kernighan's in a source level, enabling users to download and integrate it to their own system[10].

3 Information Model

3.1 Data Classification

The first step of building a recommender system is to systematically organize the large amount of data. Our model consists of two major information components, namely, TPOI information and user information. The analyzed TPOI information can be classified as shown in Fig. 1. In the figure, the largest category groups are *objective* and *subjective* attributes, respectively. While the objective attribute represents general tourist information for TPOI such as location and name, subjective attributes express information subject to the character of respective individuals such as visitor's opinion on the particular TPOI. Even with the same tourist information, each user can choose different TPOIs. Among these, the average stay time is directly exploited by the tour planner.

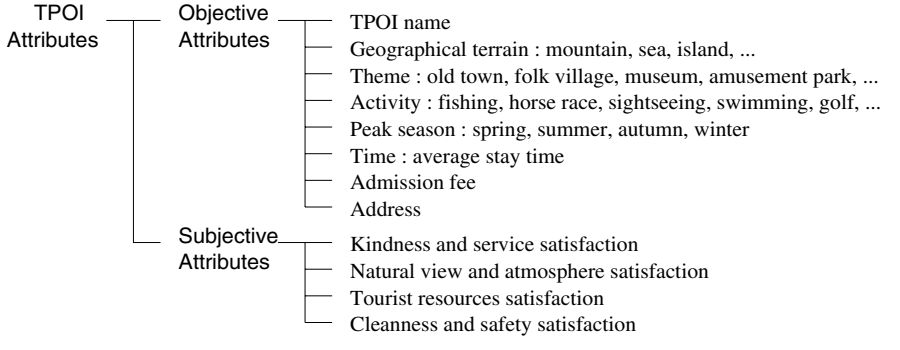


Fig. 1. TPOI information

On the other hand, the user attribute, which users need to submit to the system, contains personality, attitude, demographic, and social factors. It is further classified into *preference* attributes and *general* attributes, respectively, as shown in Fig. 2. The preference attribute describes a personal requirement on this tour. For example, one prefers sightseeing on seaside, one wants to have a night tour, and one want to experience motor sports. In addition, the general attribute describes a user's general taste to his tour. For example, demographic factors include age, gender, and income. For the sake of minimizing the input of the

user, general attributes are further classified into static and dynamic attributes according to whether the user has to specify data each time he starts another tour. Static attributes only need to be given at the initial registration phase because they don't change during the tour period. However, dynamic attributes need the user's input every time he wants a recommendation. After preference attributes are provisioned initially by the user, they are adjusted automatically with the tourist history of the user and the log-in information. A user can modify preferences at any time he wants.

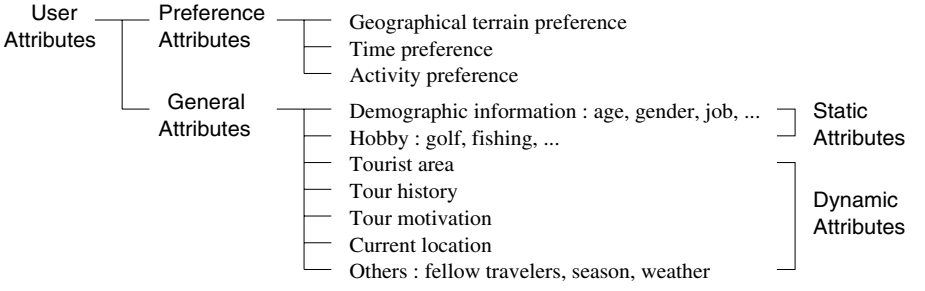


Fig. 2. User information

3.2 Computing Algorithm of the Similarity

The similarity computation scheme can be better described by an example based on the attribute classification proposed in the previous subsection. To begin with, let's assume that there are two TPOIs, namely, A and B, while individual preference attributes are given as shown in Table 1, where TPOI information is specified in terms of terrain, theme, and activity attributes. The *user preference table* entry contains priority field which ranks the degree of preference. The higher the priority value, or score, the more a user prefers.

Table 1. Given preference parameters

TPOI Information

	TPOI A	TPOI B
Terrain	Mountain	Island
Theme	Natural View	Folk Village
Activity	Golf	Sightseeing

User preference

Position	Mountain	Sea	City	Island
Priority	3	4	1	2
Theme	Folk village	Natural view	Park	Museum
Priority	4	2	1	3
Activity	Fishing	Sightseeing	Golf	Casino
Priority	4	3	2	1

With such settings, the user preference is to be compared according to the vector similarity defined in in Eq. (1), where the similarity calculation measures the cosine value between two vectors[5]. However, it has a problem when it computes the similarity between the distributed pattern of two vectors. To resolve this problem, the similarity criteria is modified to take into account both the length ratio of vectors and their size, as shown in Eq. (2).

$$VS(\overrightarrow{P_u}, \overrightarrow{O_t}) = cos\theta = \frac{\overrightarrow{P_u} \cdot \overrightarrow{O_t}}{|\overrightarrow{P_u}| \times |\overrightarrow{O_t}|} \quad (1)$$

$$VS(\overrightarrow{P_u}, \overrightarrow{O_t}) \times \frac{\overrightarrow{P_u}}{\overrightarrow{O_t}} = \frac{\overrightarrow{P_u} \cdot \overrightarrow{O_t}}{|\overrightarrow{O_t}| \times |\overrightarrow{O_t}|} \quad (2)$$

where $\overrightarrow{P_u}$ denotes the preference of user u, $\overrightarrow{O_t}$ the objective attribute of TPOI t, and $cos\theta$ the pattern similarity of u to t. Based on this similarity analysis method, the recommender system first compares the similarity between the TPOI objective attribute and user preference. If some user preference field is not given or not clear, analysis on similar users are performed to fill the undefined fields. In this step, unrelated information will be filtered out. For the more detailed description, refer to [9].

4 Recommender System

4.1 System Architecture

The overall architecture is shown in Fig. 3. To begin with, each telematics installed Microsoft Windows Mobile operating system and is connected to our system via CDMA interface using RAS (Remote Access Service) utility. Users can input their preference in the form of numbered value through the interface implemented in the telematics, the recommended result being shown also in this device. Additionally, the telematics device provides a basic map viewing functions such as zoom in-out, pan, and distance calculation based on the road network. The road network has only nodes and links, which are intersection and two end points of the road segment, respectively. In addition, along with a corresponding user interface, the device can perform A* path finding algorithm.

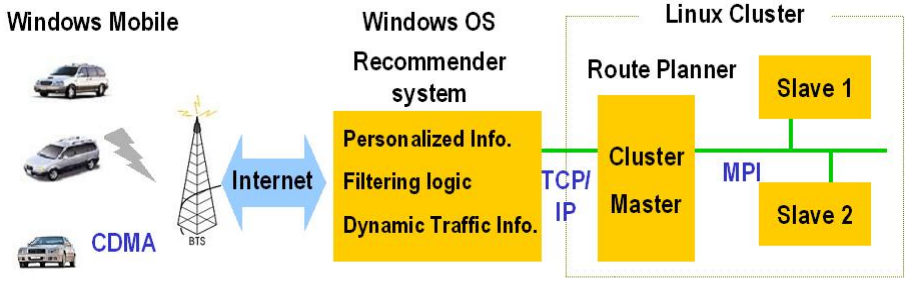


Fig. 3. System architecture

The recommender system has been built on top of general desktop Windows operating system and relational database which stores information on each personal preference and TPOI characteristics. It implements the filtering logic, as previously explained. Dynamic traffic information is available in Jeju area via the web

interface to the Jeju ITS (Intelligent Transport System) or Jeju Taxi Telematics system[1]. After the recommender system selects the candidate TPOIs, they will be handed over to the tour planning server, or route planner. The main function of tour planning server is to generate a subset of TPOIs and their tour schedule which maximizes the satisfaction degree, considering the user-specified tour duration and traffic condition. We define the satisfaction degree of a set as the sum of the score of all TPOIs in the set. The planning server mainly plays a role of TSP solver, and as this procedure needs a great deal of computation, we build a Linux cluster consist of a master and 2 slaves. But more computing nodes can be added to our system just with a minor revision of the software.

4.2 Tour Plan Generator

To begin with, we define that a TPOI set has a feasible schedule when there exists a TPOI sequence $\{S_1, S_2,..., S_n\}$ that meets the Ineq. (3).

$$\sum T(S_i) + \sum D(S_i \cdot S_{i+1}) + D(S_{i+1} \cdot S_0) \leq T_c \tag{3}$$

where $T(S_i)$ denotes the stay time at S_i, $D(S_i \cdot S_j)$ the driving time from S_i to S_j, and T_c the user-specified tour length. At the filtering step, candidate TPOIs are selected and sorted by their scores. If there exists a feasible tour schedule for the initial set, all TPOIs are selected for the final recommendation. Otherwise, the system should extract the appropriate subset that has a feasible schedule. A TPOI may be discarded if a tour schedule including it exceeds the given tour length, or if a tour schedule excluding it has a higher satisfaction degree. Furthermore, even if S_i has the higher score than S_j, S_i may be excluded when any set including S_i doesn't have a feasible schedule, while S_j has.

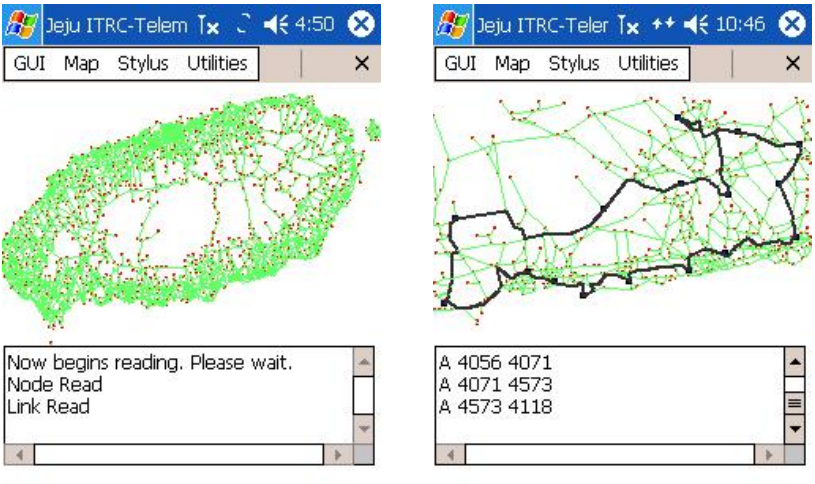
When the number of candidate TPOIs are too many, it may prolong the response time in generating a feasible schedule. Hence, it is necessary to remove the TPOIs that have little possibility to be selected for a final recommendation from the first. To this end, we select the initial subset of the TPOIs. From the sorted list, we choose TPOIs one by one until the sum of stay time is less than $1.5 \cdot T_c$. In addition, if exists, the TPOIs whose scores are same or almost same with the last TPOI will be selected to the initial TPOI set. Our experience shows that this selection rarely misses the optimal route. If the number of candidate TPOIs is n after such initialization, we should inspect whether there exists a feasible schedule for 2^n subsets. Among the feasible ones, the set of the highest satisfaction degree is finally chosen for the final recommendation. However, the time complexity of TSP is $O(n!)$, where n is the number of visiting nodes, so the total complexity of tour planning system is estimated to be $O(2^n \cdot n!)$. As natural, there are some constraints that can remove the unnecessary computation. Namely, the system doesn't have to calculate TSP for a set, either when the sum of stay time of all element already exceeds the given tour length or when the satisfaction degree is less than the current maximum.

As the TSP overwhelms the computing time of tour planning system, we need an efficient heuristic for this problem. Lin-Kernighan algorithm is known to be one of

the most efficient algorithms, while running a Lin-Kernighan algorithm needs the cost matrix having the driving time between every node pair. If we have n nodes, the cost matrix needs $n \times (n-1)$ times of A* calculations, while each execution time is dependent on the number of nodes and links in the road network. The road network of Jeju province consists of about 17,000 nodes and 30,000 links, so it takes a non-negligible time to compute this matrix but Linux cluster provides a cost-effective way to enhance the performance of such computation.

In the Linux cluster of tour planner system, the master node is equipped with 1.0 GHz Pentium IV CPU and 512 MB memory, while the slave node 700 MHz Pentium 3 CPU and 384 MB memory, and they are connected with 100 MBps Ethernet interface. In addition, the NETGEAR 8-port Fast Ethernet switch connects all of cluster nodes to build a private network. Finally, all nodes installed Redhat Linux version 9.0 and LAM-MPI version 7.1.2. The master initiates the cooperative computation by sending an MPI (Message Passing Interface) message[12]. Each node exchanges the necessary information such as candidate set, user-specified tour length, and partial result through the shared file system. If we make b_i, when set, indicate that S_i is included in the trial set, the task partition will be done as follows: The master takes the set if $b_{n-2}b_{n-1}$, namely, bits for the last two TPOIs, is 10, 11. The set of 01 and 00 will be processed at each slave. After each node calculates, the master collects the partial maximum to choose the final maximum.

Fig. 4 shows the execution result displayed in the PDA. Fig. 4(a) shows the main menu of our system interface including map viewing functions and GUI. After a tour planning system calculates a feasible schedule, it will return the tour schedule $\{S_0, S_1, ..., S_{n-1}\}$. The the system interface program will calculate A* algorithm along the received route to display the final route as in Fig. 4(b).



(a) Main menu (b) Plan generation

Fig. 4. Tour plan generation

5 Performance Measurement

Based on this system, we measured the performance of out planning system in terms of execution time, the probability of finding the optimal schedule, and the optimality of the tour schedule according to the number of candidate TPOIs as well as satisfaction degree. For the experiment, we have generated 14 set groups, each of which has 50 sets having same number of candidate TPOIs. The number of candidate TPOIs of respective groups ranges from 12 to 28. Each TPOI has its own rank distributing randomly from 1 to 10 as well as stay time exponentially distributing centered on two average, 1.0 and 0.2 for the lookout-type TPOI.

Fig. 5 plots the execution time of the proposed tour planning scheme according to the number of candidate TPOIs. This curve indicates that the tour schedule can be offered to the user within 5 seconds when the number of TPOIs is less than 22. The execution time of full search, which investigates every feasible schedule, grows beyond 1 minute, when the number of TPOIs is larger than 11. In addition, Fig. 6 shows the probability of finding optimal schedule. The probability means the ratio of the candidate sets the system finds the optimal schedule to all of the generated sets. As shown in this figure, when the number of TPOIs is less than or equal to 17, the probability of finding the optimal schedule is above 90 %.

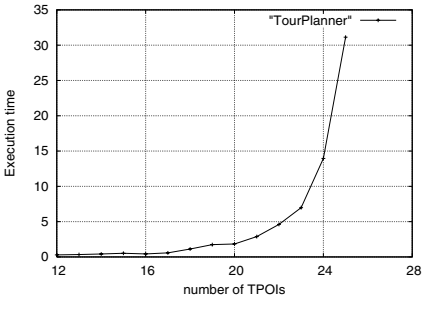


Fig. 5. Execution time measurement

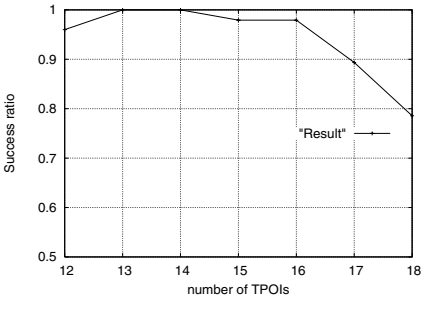


Fig. 6. Finding optimal solution

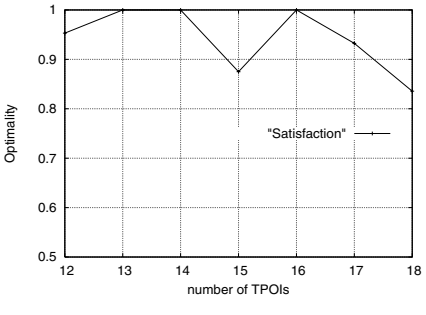


Fig. 7. Optimality vs. # of TPOIs

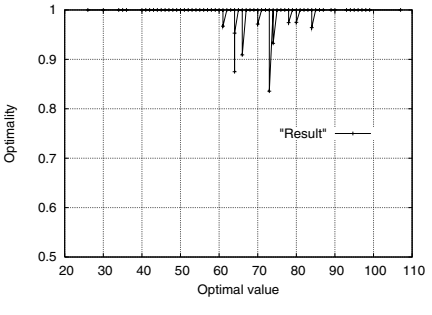


Fig. 8. Optimality vs. optimal value

Fig. 7 and Fig. 8 evaluate the quality of schedule generated by our system by measuring how close its satisfaction degree is to the optimal degree according to the number of TPOIs and optimal value. These experiments indicate that both factors have not so significant impact to the optimality when the number of TPOIs are less than 18. However, the optimality always exceeds 0.8 for those parameters.

6 Conclusion

This paper has designed and implemented an intelligent tour planning system capable of 1) selecting TPOIs based on a personalized information rather than per-TPOI preference, 2) generating a tour plan that maximizes the satisfaction degree for a tourist. To reduce the response time resulting from the calculation of $O(2^n \cdot n!)$ complexity, we used initial set reduction, distributed computing via MPI-based Linux cluster, and finally Lin-Kernighan heuristic. An user interface was also implemented on a portable device using the utility of embedded operating system. Performance measurement results show that the tour schedule can not only be offered to the user within 5 seconds when the number of TPOIs is less than 22, but also find a schedule whose satisfaction degree is close to the optimal value. We believe that this system can provide a useful information to tourists for Jeju province as a good example of telematics application. As a future work, we will keep finding a prospective application for the telematics network for the dissemination of telematics device.

References

1. Lee, J., Park, G., Kim, H., Yang, Y., Kim, P., Kim, S.: A telematics service system based on the Linux cluster. In: Shi, Y. (ed.) ICCS 2007. LNCS, vol. 4490, pp. 660–667. Springer, Heidelberg (2007)
2. Ponnada, M., Sharda, N.: A high level model for developing intelligent visual travel recommender systems, ENTER (2007)
3. Ricci, F., Werthner, H.: Case base querying for travel planning recommendation. Information Technology & Tourism 4, 215–226 (2002)
4. Maruyama, A., Shibata, N., Murata, Y., Yasumoto, K., Ito, M.: P-tour: A personal navigation system for tourism. In: Proc. of 11-th World Congress on ITS 2, pp. 18–21 (2004)
5. Breese, J., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Annual conference on Uncertainty in Artificial Intelligence, pp. 43–52 (1998)
6. Schubert, P., Koch, M.: The power of personalization: Customer collaboration and virtual communities. In: Proceedings of the Conference on AMCIS, pp. 1955-1965 (2002)
7. Nasraoui, O., Petenes, C.: An intelligent web recommendation engine based on fuzzy approximate reasoning. In Proceeding of the IEEE International Conference on Fuzzy System 2, 1116–1121 (2003)
8. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combining content-based and collaborative filters in an online newspaper. In: Proceedings of the ACM SIGIR Workshop on Recommender Systems (1999)

9. Kang, E., Kim, H., Cho, J.: Personalization method for tourist point of interest (POI) recommendation. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS (LNAI), vol. 4251, pp. 392–400. Springer, Heidelberg (2006)
10. http://www.tsp.gatech.edu/concorde.html
11. Goldberg, A., Kaplan, H., Werneck, R.: Reach for A*: Efficient point-to-point shortest path algorithms. MSR-TR-2005-132. Microsoft (2005)
12. Pacheco, P.: Parallel Programming with MPI. Morgan Kaufmann Publishers, San Francisco (1996)

Ionospheric F-Layer Critical Frequency Estimation from Digital Ionogram Analysis

Nipon Theera-Umpon

Department of Electrical Engineering, Faculty of Engineering,
Chiang Mai University, Chiang Mai 50200 Thailand
nipon@ieee.org

Abstract. The ionosphere plays an important role in radio applications, e.g., satellite, cellular phone, global positioning system (GPS), etc. An ionogram is one of the information sources of the ionosphere. Unfortunately, ionograms are generally corrupted by noise and artifacts. These imperfections cause difficulties in the efforts to create automatic systems. In this paper, we propose a size-contrast filtering-based ionogram enhancement. The size-contrast filter can suppress the objects that are too small or too large. The good enhancement performance is achieved by applying the proposed algorithm to the ionograms collected by the ionosonde at Chiang Mai University, Thailand. We also propose a critical frequency estimation algorithm for the ordinary-mode and extraordinary-mode wave components. The proposed estimation algorithm is based on the region of interest selection, and weak segment elimination and trace determination. We achieve very good estimation performance by evaluating our proposed algorithms on ionograms from 2 sets of 12 consecutive sweeps from the nighttime and daytime.

Keywords: Ionogram, Size-contrast filter, Critical frequency, Ordinary-mode wave, Extraordinary-mode wave.

1 Introduction

The ionosphere is the earth's atmospheric region in which particles are ionized. It ranges from the altitude of about 60 km to 1000 km. It plays an important role in radio communications. The change in the ionosphere causes the problems in our communications, for example, radio absorption, interference, scintillation, propagation delay, etc. The problems affect many radio applications including cellular phone, satellite communications, TV and radio broadcasting, global positioning system (GPS), etc. It has been known that the ionosphere can reflect radio waves in the high frequency (HF) band (about 3 to 30 MHz) due to the ionized particles. Because of the dynamic nature of the ionosphere, we need to understand its behavior and characteristics.

To study such ionospheric characteristics, an ionosonde station transmits and receives HF radio waves via a wired antenna. The plot of the sounding frequency versus the time delay of the echo signals reflected from the ionosphere or versus the height is called an *ionogram*. Unfortunately, ionograms are generally corrupted by

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, pp. 190–200, 2007.
© Springer-Verlag Berlin Heidelberg 2007

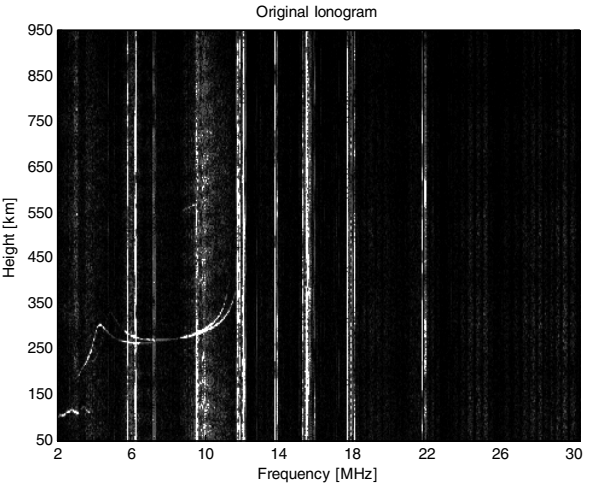


Fig. 1. A sample ionogram collected by the ionosonde at Chiang Mai University, Thailand

high level noise presenting in the HF band [1]. They are shown as the vertical bands in ionograms. These bands are caused by fixed-frequency radio transmitters. The elimination or reduction of these bands while preserving the ionospheric characteristics in the ionograms is desirable before we inspect them manually or use algorithms to extract useful information from them automatically. Because the ionogram inspection by a human expert is tedious, an automatic approach is preferable. A sample ionogram collected by the ionosonde station at Chiang Mai University is shown in Figure 1.

Many efforts have been done to automatically extract ionospheric parameters. In previous research, ionogram enhancement was based on median filtering and edge detection [1,2]. Filtering, thinning, trimming, fitting, and merging were employed in [3]. Different types of windows were also applied during the ionogram generation process [1,4]. New trimming algorithm was also proposed for this purpose [5]. The critical frequency prediction is another popular issue in this research area [6–8]. A new technique was proposed in [9] to describe the earth-ionosphere waveguide. The effects of the ionosphere artifacts in synthetic aperture radar systems were also studied [10]. Automatic approach to extract the critical frequencies in the F-layer of the ionosphere was proposed in [1]. However, their ionograms were much cleaner than what we have in our experiments and there was no comparison to any ground truths.

In this research, we propose a new technique to enhance ionograms by reducing noise and artifacts using our previously proposed size-contrast filter. The enhanced ionograms provide clearer traces. This is useful for further manual or automatic inspection. To prove that the validity of the enhanced ionograms, we propose another new technique to extract ionospheric critical frequencies for the ordinary-mode and extraordinary-mode wave components from the ionograms.

This paper is organized as follows. In section 2, we describe the details of the ionogram data used in our experiments. The proposed techniques including the

size-contrast-filtering-based ionogram enhancement and an automatic algorithm for critical frequency estimation are given in section 3. The experimental results are shown and discussed in section 4. Finally, we conclude the paper in section 5.

2 Ionogram Data Set

The ionograms used in the experiments were collected at Chiang Mai University (18.8°N, 98.9°E) by the PCS4 FMCW ionosonde consisting of SKI-09085 controller/dual channel receiver with 2-30 MHz observation frequency, A32-20LPF 20W power amplifier and SKI-01082 low pass filter bank and peripherals. The observations of the ionosonde were scheduled to collect every 5 minutes. This ionosonde station is a part of the Southeast Asia Low-latitude Ionospheric Network (SEALION.)

A sample ionogram image in this data set is as shown in Figure 1. It was collected at 16:00 hours standard Thailand time on February 20, 2007. It should be noted that, in the discrete domain, the image size is 451×700. The values on the axes can be easily converted to the frequency (1 pixel or 1 frequency bin = 40 kHz) and height (1 pixel or 1 height bin = 2 km) as shown in Figure 1. In this paper, we will present images in terms of both pixels and frequency and height.

3 Proposed Techniques

3.1 Ionogram Enhancement Using Size-Contrast Filtering

We previously proposed the following size-contrast filter in [11] to detect buried land mines of a particular range of sizes in ground penetrating radar (GPR) images. It was applied to discriminate the size of land mines and other non-mine objects that have smaller or bigger sizes than that of land mines. The results showed that it could reduce false alarms dramatically. The size-contrast filter is defined by

$$\mathbf{X}_{scf}(i,j) = \text{median}\left(inX(i,j)\right) - \text{median}\left(outX(i,j)\right), \tag{1}$$

where $inX(i,j) = \{\mathbf{X}(i,j-in), \mathbf{X}(i,j-in+1), \ldots, \mathbf{X}(i,j+in)\}$ and $outX(i,j) = \{\mathbf{X}(i,j-bound), \ldots, \mathbf{X}(i,j-out), \mathbf{X}(i,j+out), \ldots, \mathbf{X}(i,j+bound)\}$. $\mathbf{X}$ and $\mathbf{X}_{scf}$ are input and output images, respectively. The variables *in*, *out*, and *bound* are shown in Figure 2. The output value from the size-contrast filter will be high when the width of a peak is roughly between 2*in* and 2*out*. Therefore, we can filter out objects that are too small or too large.

In this particular problem, we apply the size-contrast filtering in the height axis (y-axis) of ionograms. In the experiments, we set *in*=1, *out*=3, and *bound*=5. Because the traces near the critical frequencies are weak compared to other parts, we normalize each column of the ionograms by its maximum value, i.e.,

$$\mathbf{X}_{norm}(i,j) = \mathbf{X}_{scf}(i,j) / \max\{\mathbf{X}_{scf}(j)\}, \tag{2}$$

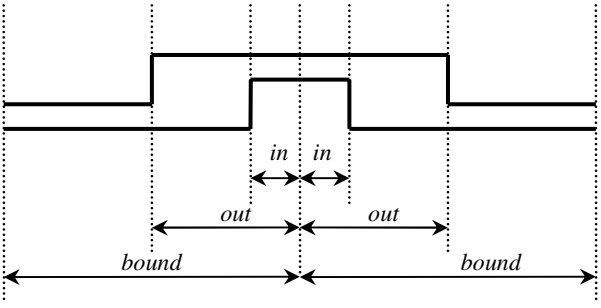


Fig. 2. In, out, and bound variables in the size-contrast filtering

for $i = 1$ to the number of rows and $j = 1$ to the number of columns, and $\mathbf{X}_{scf}(j)$ is the j^{th} column of the size-contrast filtered image. This will enhance the traces notably and will make the critical frequency estimation step easier.

3.2 Critical Frequency Estimation

The enhanced ionograms are further employed to estimate the critical frequencies of the F-layer ionosphere. This proposed estimation technique includes two main parts, i.e., region of interest selection and weak segment elimination and trace determination.

3.2.1 Region of Interest Selection

To avoid noise and artifacts in other regions of ionograms, a region of interest selection technique is proposed. The idea is to select the area in ionograms covering only the F-layer of the ionosphere. From the enhanced ionograms, we notice that the strong traces occur in F-layer. We find the energy at each height and choose the height where the energy peak occurs to represent the trace location, i.e.,

$$\text{Trace Location} = \arg\max_{i} \sum_{j=1}^{700} \mathbf{X}_{norm}^{2}(i, j) \,. \tag{3}$$

We set the region of interest to cover the height ranging from 40 km below the trace location to 200 km above the trace location to ensure the F-layer coverage.

3.2.2 Weak Segment Elimination and Trace Determination

Considering the derived region of interest, we need to specify the segments of traces. The candidate pixels for the traces are found by choosing two pixels with the highest levels in each column. This is because we usually have two traces for the ordinary-mode and extraordinary-mode wave components. For each frequency bin j (j^{th} column) we label the two candidate pixels $p_1(j)$ and $p_2(j)$.

The above method creates a lot of pixels that are not related to the traces. We eliminate the weak segments based on the idea that the strong traces should have

elongated clusters of pixels. To be more specific, we consider the following conditions for each candidate pixel $p_k(j)$:

$$\min\{|p_k(j+1) - p_1(j)|, |p_k(j+1) - p_2(j)|\} < b$$
$$\min\{|p_k(j+2) - p_1(j)|, |p_k(j+2) - p_2(j)|\} < b$$
$$\min\{|p_k(j+3) - p_1(j)|, |p_k(j+3) - p_2(j)|\} < b$$
$$\min\{|p_k(j+4) - p_1(j)|, |p_k(j+4) - p_2(j)|\} < b$$
$$\min\{|p_k(j+2) - p_1(j+1)|, |p_k(j+2) - p_2(j+1)|\} < b$$

The decision rule is as follows:

If the number of the above conditions that are true are greater than two, then we declare that the pixel $p_k(j)$ is a part of strong traces.

If not, the pixel $p_k(j)$ is a part of a weak segment.

It is worthwhile noting that we have two candidate pixels per frequency bin, so, for each j, we have to perform the decision making twice for $k = 1$ and 2. It should also be noted that b is a preset parameter. The value of b depends upon the high slope of the end of traces near critical frequencies. In the experiments, we set b to 15 which implied to the slope of about 86 degrees.

After the weak segment elimination, we will have only strong traces left. For each frequency bin, there may be none, one, or two pixels. The traces for the ordinary-mode and extraordinary-mode wave components are determined by these following rules:

For each frequency bin

If there is no pixel, then there is no trace in that frequency bin.

If there is one pixel, then it is a part of trace for the extraordinary-mode wave components..

If there are two pixels, then the upper pixel is a part of trace for the ordinary-mode wave components, and the lower pixel is a part of trace for the extraordinary-mode wave components.

The critical frequencies can be easily determined from the location of the right-end of the traces.

4 Experimental Results and Discussion

We apply the proposed enhancement technique to the ionogram data set described in section 2. The size-contrast filtered image of the ionogram shown in Figure 1 is shown in Figure 3. It can be seen that the image is clearer than the original image. Most parts of the noise bands are eliminated. However, the traces are still too weak, especially near the critical frequencies. The traces are more visible in the normalized image shown in Figure 4. One problem we can see here is the missing parts of the traces due to the bands of noise. This is extremely difficult to avoid because the noise levels are generally higher than the signal levels in the traces.

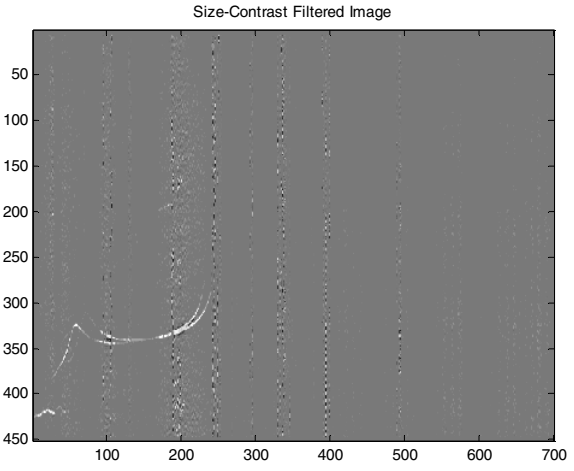


Fig. 3. Size-contrast filtered image of the ionogram shown in Figure 1

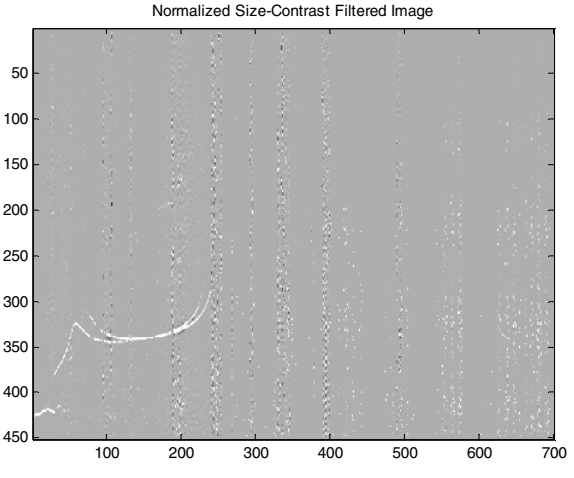


Fig. 4. Enhanced image of the ionogram shown in Figure 1

To determine the region of interest, we compute the energy at each height bin as shown in Figure 5. From the plot, the height where the energy peak occurs is at the height bin 343. Therefore, we select the height bins 263 to 363. The candidate pixels in the region of interest are shown in Figure 6. We can see that the image is very noisy. After the weak segment elimination step, however, we end up with a much cleaner image with traces as shown in Figure 7. The final result in Figure 8 depicts the original ionogram superimposed with the extracted critical frequencies for the ordinary-mode wave component (f_oF_2) and for the extraordinary-mode wave component (f_xF_2). In this particular example, the estimated critical frequencies for the ordinary- and extraordinary-mode wave components are 11.04 and 11.60 MHz, respectively.

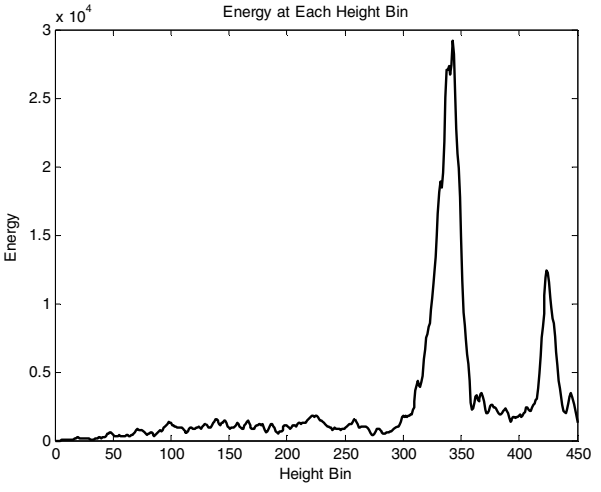


Fig. 5. Energy at each height bin of the enhanced ionogram shown in Figure 4

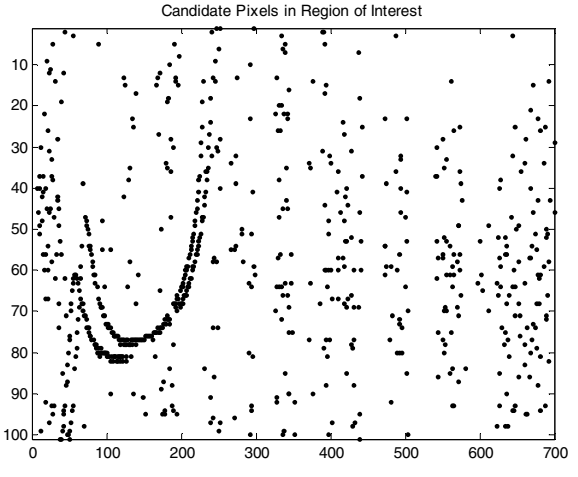


Fig. 6. Candidate pixels of the ionogram shown in Figure 1

We consider another example of ionogram collected in the nighttime (02:00 hours) on February 20, 2007. In contrast to the ionogram shown in Figure 1 which was collected in the daytime (16:00 hours), this ionogram does not show some bands of noise (some radio stations did not broadcast at night.) However, we can see that the noise bands in the lower frequencies are more visible. Figure 9 shows the original ionogram. The estimated critical frequencies, $f_oF_2 = 4.72$ and $f_xF_2 = 5.40$, superimposed on the original image are shown in Figure 10. We can see that the critical frequencies for both modes of wave components are lower than that during the daytime.

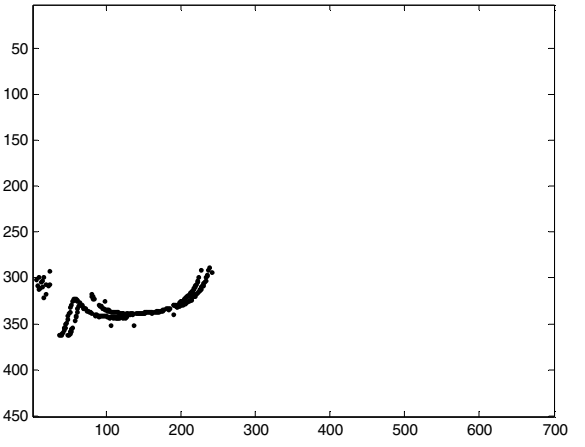


Fig. 7. Traces of the ionogram shown in Figure 1

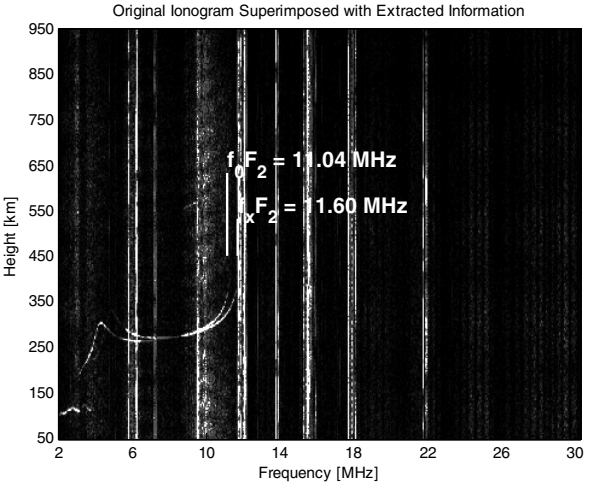


Fig. 8. Ionogram in Figure 1 superimposed with the estimated critical frequencies

To further evaluate the proposed algorithm's performance, we compare the algorithm's outputs to the critical frequencies estimated from ionograms by a human. Due to the tediousness for humans to estimate the critical frequencies manually, we randomly perform the estimation by human visualization for 12 consecutive sweeps in the nighttime, i.e., from 00:00 to 00:55 hours standard Thailand time on February 20, 2007. The results are shown in Table 1. The average percentage errors of our critical frequency estimation algorithm for the ordinary-mode and extraordinary-mode wave components are 2.46 % and 3.55 %, respectively. We can see that our estimates are very close to the human opinions. We also show the results from the daytime, i.e.,

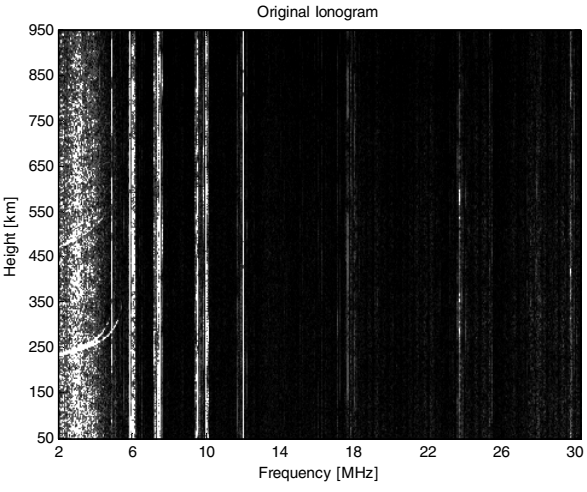


Fig. 9. Ionogram collected in the nighttime by the ionosonde at Chiang Mai University

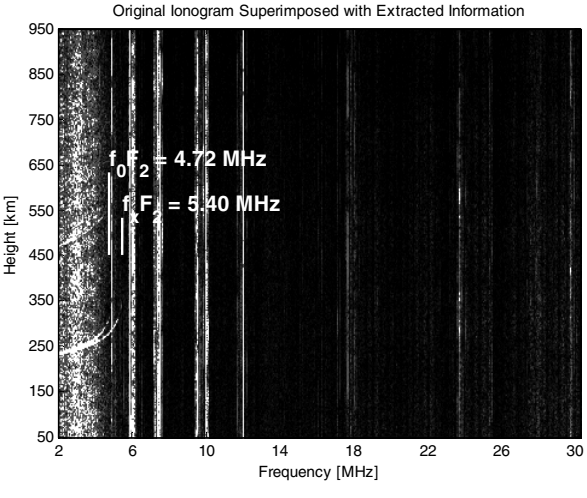


Fig. 10. Ionogram in Figure 9 superimposed with the estimated critical frequencies.

from 15:00 to 15:55 hours standard Thailand time on February 20, 2007 in Table 2. The average percentage errors of our critical frequency estimation algorithm for the ordinary-mode and extraordinary-mode wave components are 1.08 % and 1.13 %, respectively. The estimation from the ionograms collected in the daytime is better than that collected in the nighttime. From Figures 1 and 9, we can see that, in the nighttime, the noise bands in the low frequencies appear in the range where the traces occur. Therefore, they cause difficulties in the critical frequency estimation.

Table 1. Critical frequency estimation by the proposed algorithm compared to that from human visualization (from 00:00 to 00:55 hours standard Thailand time on February 20, 2007)

Ionogram Frame Index	Critical Frequency [MHz]			
	O-Mode Wave Component		X-Mode Wave Component	
	Human Visualization	Proposed Algorithm's Estimate	Human Visualization	Proposed Algorithm's Estimate
1	4.80	4.64	5.32	5.04
2	4.80	4.72	5.40	5.44
3	4.76	4.68	5.36	5.16
4	4.68	4.40	5.32	5.12
5	4.68	4.72	5.20	5.20
6	4.76	4.72	5.40	5.20
7	4.76	4.64	5.44	5.04
8	4.80	4.56	5.40	5.16
9	4.76	4.64	5.36	5.20
10	4.64	4.56	5.40	5.32
11	4.68	4.60	5.32	4.96
12	4.76	4.68	5.36	5.24
Average Percentage Error	2.46 %		3.55 %	

Table 2. Critical frequency estimation by the proposed algorithm compared to that from human visualization (from 15:00 to 15:55 hours standard Thailand time on February 20, 2007)

Ionogram Frame Index	Critical Frequency [MHz]			
	O-Mode Wave Component		X-Mode Wave Component	
	Human Visualization	Proposed Algorithm's Estimate	Human Visualization	Proposed Algorithm's Estimate
1	11.08	10.92	11.44	11.56
2	11.20	11.04	11.48	11.80
3	11.16	11.08	11.52	11.36
4	11.20	11.08	11.52	11.84
5	11.16	11.12	11.48	11.48
6	11.12	10.96	11.48	11.36
7	11.20	11.12	11.60	11.72
8	11.20	11.08	11.52	11.60
9	11.16	11.00	11.52	11.56
10	11.12	11.04	11.56	11.52
11	11.16	11.40	11.56	11.56
12	11.08	11.04	11.52	11.76
Average Percentage Error	1.08 %		1.13 %	

5 Conclusion

In this paper, we propose a size-contrast filtering-based algorithm to enhance ionograms. The size-contrast filter can suppress the objects that are too small or too large. It is applied in the height axis of ionograms to filter out the noise bands (too large) and speckle noise (too small.) By applying the proposed algorithm to the ionograms

collected by the ionosonde at Chiang Mai University, it yields promising results. The enhanced ionograms are further applied in the critical frequency estimation for the ordinary-mode and extraordinary-mode wave components. The proposed estimation algorithm is based on the region of interest selection, and weak segment elimination and trace determination. We evaluate our proposed algorithms by estimating the critical frequencies of ionograms from 2 sets of 12 consecutive sweeps from the nighttime and daytime collection. The low average percentage errors suggest that we achieve very good estimation performance.

Acknowledgment

This work is partially supported by the National Institute of Information and Communications Technology (NICT), Japan, under the Southeast Asia Low-latitude Ionospheric Network (SEALION) program. The author would like to thank Dr. Takashi Maruyama (NICT) for his contribution through our technical discussion on this research.

References

1. Arikan, F., Arikan, O., Salous, S.: A New Algorithm for High-Quality Ionogram Generation and Analysis. In: Radio Science 37(1), 4-1–4-11 (2002)
2. Arikan, F., Salous, S., Arikan, O.: A New Algorithm for High Quality Ionograms. Electronics Letters 36(11), 985–987 (2000)
3. Redding, N.J.: Image Understanding of Oblique Ionograms: The Autoscaling Problem. In: Proc. 1996 Australian New Zealand Conf. on Intelligent Information Systems (1996)
4. Barlett, A., Gallagher, M., Darnell, M.: Extraction, Analysis and Interpretation of Digital Ionogram. In: Proc. 6th IEEE Intl. Conf. on HF Radio Systems and Techniques, pp. 278–282 (1994)
5. Kettler, D.I., Redding, N.J.: A Trimming Algorithm to Clean Thinned Feature for Feature Extraction in Image Understanding. In: Proc. 1996 Australian New Zealand Conf. on Intelligent Information Systems, pp. 304–307 (1996)
6. Zain, A.F.M., Abdullah M.: Application of Artificial Intelligence to Ionospheric Radio Propagation. In: Proc. IEEE TENCON. vol. 1, pp. 82–87 (2000)
7. Chan, A., Cannon, P.S.: Degradation in Neural Network Prediction Models of f0F2 with Time. In: Proc. IEE Intl. Conf. on Antennas and Propagation. vol. 2, pp. 787–791 (2001)
8. Cander, L.R., Bamford, R., Hickford, J.G.: Nowcasting and Forecasting the foF2, MUF(3000)F2 and TEC Based on Empirical Models and Real-Time Data. In: Proc. Intl. Conf. on Antennas and Propagation. vol. 1, pp. 139–142 (2003)
9. Roostaei, S., Sadat, S., Ghobadi, C., Nourinia, J.: Modeling and Numerical Analysis of the Earth-Ionosphere Waveguide using Finite Difference Method. In: Proc. Intl. Symp. on Antennas and Propagation Society. vol. 2B, pp. 158–161 (2005)
10. Chapin, E., Chan, S.F., Chapman, B.D., Chen, C.W., Martin, J.M., Michel, T.R., Muellerschoen, R.J., Pi, X., Rosen, P.A.: Impact of the Ionosphere on an L-band Space Based Radar. In: Proc. IEEE Conf. on Radar, pp. 51–58 (2006)
11. Theera-Umpon, N., Gader, P.D., Ho, D.K.: Region-Based Time-Domain Processing of Hand-Held GPR. In: UXO Countermine Forum (2001)

Study of Digital License Search for Intellectual Property Rights of S/W Source Code

Byungrae Cha[1], Kyungjun Kim[2,⋆], and Dongseob Lee[3]

[1] Dept. of Computer Eng.,
[2] Dept. of Radio Mobile Communication Eng.,
kjkim@honam.ac.kr
[3] Dept. of Information and Communication Eng.,
Honam Univ., Korea
{chabr, kjkim, dslee}@honam.ac.kr

Abstract. The IPR(Intellectual Property Rights) system and technology for the extended software source code from digital contents are very significant to enhance international competitiveness. A problem happens if the original software source code is made public at dispute on its ownership. This study has suggested the Business model and comparison of S/W source code vs. Digital License to protect the copyright infringement and technique leakage caused by its opening. In this paper, we make a study of the digital licence prototype for discriminate the original source code. Reserved words of software source code by parsing express to XML file that have hierarchical structure. Then, we can express architecture of software source code by tree structure form instead of complex source code. And we make a study of the indexing and searching to search digital license.

1 Introduction

Traditionally the copyright has been handled and studied from the point of legal views. The topics of the study include the nature of copyright, the scope of its protection, execution and infringement as follows. Firstly, the copyright is a legal system to provide a safe and stable environment for various creative original works. There are several grounds for this rising economic value of copyright. Above all, the concerns on the copyright result from the growing recognition on the enforced role of Intellectual Property(IP) in the knowledge-based society which pays further attention to unmaterial factors. The copyright is significant for the growth, productivity, employment and investment of creativity- and information-based industry. Secondly, the scope of copyright protection is largely broadened in pace with the development of digital technology. The economic benefits from software, multimedia and other technical outcomes get enormous. Finally, the objects of copyright protection become crucial elements of electronic commerce as a result of digital revolution. The system of Intellectual Property

⋆ Corresponding Author

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 201–212, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Rights(IPR) has played a crucial role for the progress of information society in the 21st century as it did for the birth and development of previous industrial society. In link to the source code of extended software in digital contents, the well-organized IPR system and description of rights have very important meaning in Korea as well with a view to enhancing international competitiveness. The management technology of copyright of software source code stays at beginning stage, compared with the studies on the copyright protection of digital contents. The problem comes from opening the software source code to prove the ownership if a dispute for software source code happens. This study is purposed to suggest digital license prototypes and business models for software source code in order to prevent the copyright infringement and the leakage of technique after publicizing the original software source code. The suggested digital license for software source code includes the information on the developer and structural information of source code. Accordingly the information on the developer's ownership and the structural information of source code enable to judge the originality of software on dispute without providing the source code firsthand. And we make a study of the indexing and searching to search digital license.

2 Related Works

2.1 DRM Technology

DRM technology[1 - 6] is a management system to protect the proper copyright of digital contents produced on various purposes during the production, the distribution and the utilization. DRM technology can be categorized into two section largely. One is involved in the copyright management, and the other is in the copyright protection. The technique of the copyright protection takes up the majority of commercial DRM technology at the moment. It is understood as the technique to force a series of principles and scenarios defined in the technique of copyright management. The technique of the copyright management is to establish a globally unified management system for a set of digital works. Many organizations drive to standardize the description language for contents identifier and contents meta data. And Study of pattern matching of Software source code was studied by Rieger[7] and Yang[8]. Rieger achieved study that software source code detects visualizing part that is duplicated or is reproduced, Two different programs study that find equal code as syntactic achieve by Yang. We have gone SIM[9], Dup[10], Plague, YAP[11], YAP3, MOSS[12] in abroad, and clonechecker of KAIST, LOFC of Busan Univ., exEyesLight[13] in domestic for program plagiarism detection S/W. This study is carried out to support to DRM for software source code. The DRM description for software source code is not available. The encryption technology of IPR protection technology takes an optional effect on 1:1 trade, which stays at the initial stage. The software source code has a meaning far above a plant's design sheet, and it is an important resource of digital contents as much as production line. But the DRM technology to support this resource falls far behind.

2.2 Necessity of Source Code Protection Technology and Business Model

It is assumed that a suit for illegal leakage of software source code is instituted among software developers A, B and C at court D (Fig. 1). The dispute occurs between A, owner of original source code, and B, illegal leaker (Fig. 1, (1)). The court orders developer A and B to submit their source code for the settlement(Fig. 1, (2)). The software source code needs to be inspected by an expert, and the court D inquires the differentiation between original and leaked code to special software developer C(Fig. 1, (3)). Special software developer C has to analyze the source code to compare the original and leaked code. The differentiation between original and leaked code depends on the C's opinions completely, so the objectivity is deficient(Fig. 1, (4)). Further, it is secondary source code leakage(Fig. 1, (5)) and unintentional infringement of the IPR for source code. That is, the source code has to be exposed automatically in the middle of clarifying the differentiation between original and leaked one, which may bring about secondary disputes.

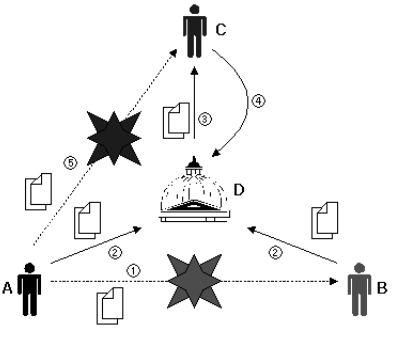


Fig. 1. The problem in the secondary IPR infringement

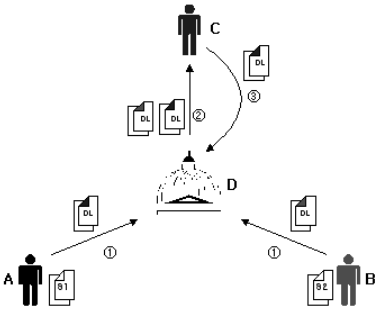


Fig. 2. IPR infringement Protection by Digital License

The problem in the IPR infringement(Fig. 1) can be ruled out through the application of digital license (Fig. 2). The source code is not informed to the court at the dispute between A and B, but it is generated in digital license and furnished to the court instead(Fig. 2, (1)). The court can obtain objective report on the matching degree via pattern matching program and get additional advice from the developer C for the decision(Fig. 2, (2)). Using digital license may help removing the apple of the secondary dispute. The developer C shall verify the digital license instead of source code and give advice how much the leaked software source code is matched with the original one(Fig. 2, (3)).

The business model using digital license[17] will protect the A's IPR, facilitate the identification of the source code leaker B and prevent a secondary potential dispute on unintentional leakage. Besides, the decision can base on the objective

and obvious report from pattern matching program, not on C's subjective views. It is not mainly affected by the C's advice after long-lasting analysis, and so the settlement of the dispute shall be shortened thanks to quick analysis of the pattern matching program.

3 Design of Digital License of Software Source Code

3.1 Analyze the Structure of Source Code

A lexical analyzer of compiler can be adopted to analyze the structure of program source code. The compiler's path tree is replaced with DOM tree which enables to use neutral language in this study, while lexically analyzed tokens are generated as path tree in tree generator and used for code generator in complier. The software source code can be viewed as a kind of systemized file like XML. Therefore DOM model, parsing outcome of XML file, is able to be used for modelling the structure of source code as well. This study is also to propose a building technique of reserved words, library and function as DOM tree via parsing of software source code (Fig. 3). 32 reserved words are defined in ANSI standard C programming language. Many C compilers add other reserved words to support the efficiency of compiler environment, joint programming with other language, interrupt and memory maintenance. The programs in C programming language are composed of main() function and subfunctions. In consequence, main() function becomes the root node of software architecture DOM tree of digital license, and subfunctions shall consist of child node of main() function. Each node has the data of library and parameters and pattern information of operators.

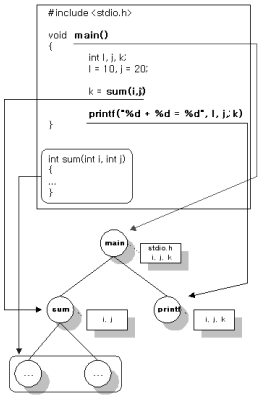


Fig. 3. Transformation Process to Software Architecture DOM Tree from Source Code

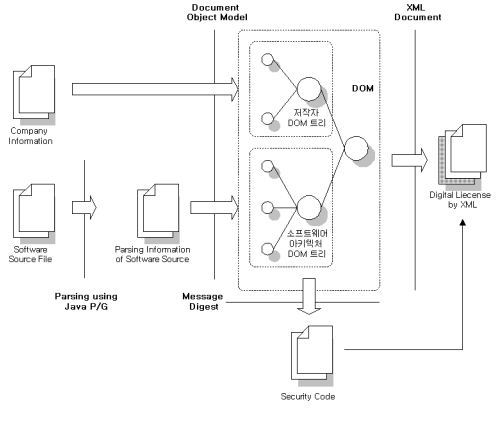


Fig. 4. Digital License Generation Process of Software Source Code

3.2 Generation Process of Source Code Digital License

The digital license of software source code has the root node called software digital license(SWDL) and two child nodes under it. The child node is constituted of author DOM tree for the information on the author of software source code and software architecture DOM tree indicating the architecture of software source code as shown in the fig. 4. Message Digest(MD)[14] is added to author DOM tree to insure the integrity of contents and information of general software registration. The core of digital license of software source code is the software architecture DOM tree. This child node is a tree- formed architecture of reserved token and software structure generated through parsing of software source code as shown in the fig. 3.

3.3 DOM Child Node Generation

The root of software architecture node undertakes the main() function. Several child nodes are allocated under the root node of main() function. The child node consists of the code block, Control Statement and function call of software source code. The software source code comprises the block marked in brace ({, }). The block is generated to child node as code package to be executed. The block may comprise the subfunction, the command sentence to be executed subsequently and the block. The block is created as child node. The control statement of programming language is classified into conditional statement, iterative statement and branch statement. In the control statement, the programming jobs' order or procedure is altered under given conditions. The implementation of program shall differ from given conditions even for the execution of the same program. In this case, the control statement is generated as child node. C programming language is composed of more than 1 function. It is largely divided into main() function and subfunction and generates the subfunction as child node. The subfunction can call the other subfunctions and itself to create them as child node.

3.4 Pattern Generation of Operator and Input/Output Data Type in Child Node

The child node is not made in the absence of operator of function code. The function with operator is able to generate new output corresponding to the input. In other words, the function with operator is thought as a processing, while the code without operator is merely a display on the screen, or it is considered as blank code to disguise the original code in ill-intended case. Therefore, the tree digest is required to cope with this code. Each node-specialized pattern is to be produced. The pattern shown in the node is to create the node pattern in internal format, using the input data type and number, operator and output data type and number. If the pattern data of node is applied, the difference in the functions of program modules can be identified via the node's pattern data, even though the software architectures are same.

4 Indexer and Search Engine of Digital License

4.1 Index Segments and Incremental Indexing

In this study, Indexer using Lucene[16] uses a well-known index structure called an inverted index. An inverted index is an inside-out arrangement of documents(digital licenses of S/W source code) such that terms(reserved words) take center stage. Each reserved word refers to the digital licenses that contain it. A Lucene index consists of one or more segments, and each segment is made up of several index files. Index files that belong to the same segment share a common prefix and differ in the suffix. We can think of a segment as a subindex, although each segment isn't a fully independent index. Each segment contains one or more Lucene Documents, the same ones we add to the index with the addDocument(Document of digital license) method in the IndexWriter class.

Using segments lets we quickly add new digital licenses to the index by adding them to newly created index segments and only periodically merging them with other, existing segments. This process makes additions efficient because it minimizes physical index modifications. One of Lucenes strengths is that it supports incremental indexing, which isnt something every IR library is capable of. Whereas some IR libraries need to reindex the whole corpus when new data is added to their index, Lucene does not. After a digital license has been added to an index, its content is immediately made searchable. In IR terminology, this important feature is called incremental indexing. The fact that Lucene supports incremental indexing makes Lucene suitable for environments that deal with large bodies of information where complete reindexing would be unwieldy. Because new segments are created as new digital licenses are indexed, the number of segments, and hence index files, varies while indexing is in progress. Once an index is fully built, the number of index files and segments remains steady.

4.2 Indexer and Search Engine of Digital License

Indexer and Search engine for digital license were embodied using Lucene API. Lucene is a high performance, scalable Information Retrieval(IR) library. To search large amounts of text quickly, we must first index that text and convert it into a format that will let we search it rapidly, eliminating the slow sequential scanning process. This conversion process is called indexing, and its output is called an index. An index as a data structure allows fast random access to words stored inside it. Searching is the process of looking up words in an index to find documents where they appear. The quality of a search is typically described using precision and recall metrics. Recall measures how well the search system finds relevant documents, whereas precision measures how well the system filters out the irrelevant documents. A Term is the basic unit for searching. Similar to the Field object, it consists of a pair of string elements: the name of the field and the value of that field. Note that Term objects are also involved in the indexing process. Digital license indexer using lucene stores the input in a data structure known as an inverted index. This data structure makes efficient use of disk space while allowing quick keyword lookups. What makes this

structure inverted is that it uses tokens extracted from input digital licenses as lookup keys instead of treating digital licenses as the central entities. The rationale for the index structure is two-fold: maximum performance and minimum resource utilization. Digital license indexer's index structure is a case study in itself of highly efficient data structures and clever arrangement to maximize performance and minimize resource usage. Digital license indexer using lucene supports two index structures: multifile indexes and compound indexes. Digital license indexer's index consists of one or more segments, and each segment is made up of several index files. Index files that belong to the same segment share a common prefix and differ in the suffix. Using segments lets you quickly add new digital licenses to the index by adding them to newly created index segments and only periodically merging them with other, existing segments. This process makes additions efficient because it minimizes physical index modifications. One of digital license indexer's strengths is that it supports incremental indexing, which isn't something every IR library is capable of. Whereas some IR libraries need to reindex the whole corpus when new data is added to their index, digital license indexer using lucene does not. After a digital license has been added to an index, its content is immediately made searchable. In IR terminology, this important feature is called incremental indexing. The fact that digital license indexer using lucene supports incremental indexing makes digital license indexer suitable for environments that deal with large bodies of information where complete reindexing would be unwieldy. Because the compound-file index structure is the default of digital license indexer, we disable it and switch to a multifile index. Although the multifile index structure is straightforward and works for most scenarios, it isn't suitable for environments with large number of indexes, indexes with a large number of fields, and other environment where using lucene results in a large number of index files. Instead of having to open and read many files from the index, as in the multifile index, Digital License Indexer using lucene must open only two files when accessing this compound index, thereby consuming fewer system resources. The compound index reduces the number of index files, but the concept of segments, documents, fields, and terms still applies. The difference is that a compound index contains a single .cfs file per segment, whereas each segment in a multifile index contains consists of seven different files. The compound structure encapsulates individual index files in a single .cfs file. Digital license indexer using lucene uses a well-known index structure called an inverted index. Quite simply, and probably unsurprisingly, an inverted index is an inside-out arrangement of documents such that terms take center stage. Each term refers to the digital licenses that contain it. File structure of inverted index was consisted of field names(.fnm), term dictionary(.tis), term frequencies(.frq) and term positions(.prx). The .fnm file contains all the field names used by digital licenses in the associated segment. Each field is flagged to indicate whether it's indexed or vectored. The order of the field names in the .fnm file is determined during indexing and isn't necessarily alphabetical. All terms (tuples of field name and value) in a segment are stored in the .tis file. Terms are ordered first alphabetically by field name and then by value within

a field. Each term entry contains its digital license frequency: the number of digital licenses that contain this term within the segment. Term frequencies in each document are listed in the .frq file. The frequency of a term in a digital license factors into the score calculation and typically boosts a digital license's relevance for higher frequencies. For each digital license listed in the .frq file, the positions (.prx) file contains entries for each occurrence of the term within a digital license. The .prx file lists the position of each term within a digital license. The position information is used when queries demand it, such as phrase queries and span queries. Position information for tokenized fields comes directly from the token position increments designated during analysis.

5 Simulation of Digital License

5.1 Comparison of Digital License Based on XML vs. S/W Source Code

The digital license of software source code gives the information on software writer, programming language, registration date, registration number, version and DOM information of software source code in regards to XML-based software source code. Parts of XML-based digital license code needs to be encrypted for the integrity and confidentiality of digital license of software source code.[15] In this study, the message digest is implemented via JCE[14] package as shown in the fig. 5. And, Fig. 6 shows search procedure of Digital License. Search procedure was consisted of 3 steps : Index Pattern, Architectur Pattern and Node Pattern.

The result of file size comparisons for 160 source code samples are shown in fig. 7 and fig. 8. In fig. 7, digital license of occasion that size of source code is very small turned on better 6 times than size of source code. Overhead increased

Fig. 5. Digital License Prototyping of Software Source Code

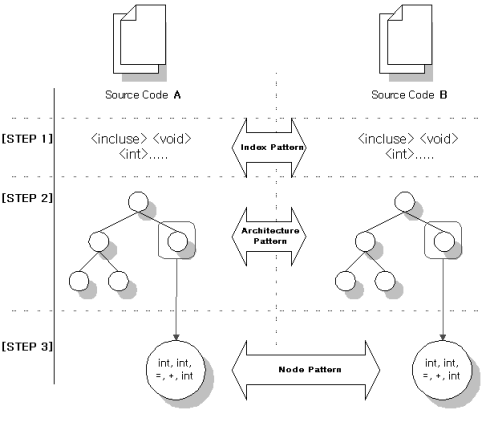


Fig. 6. Searching 3 Steps of Digital License

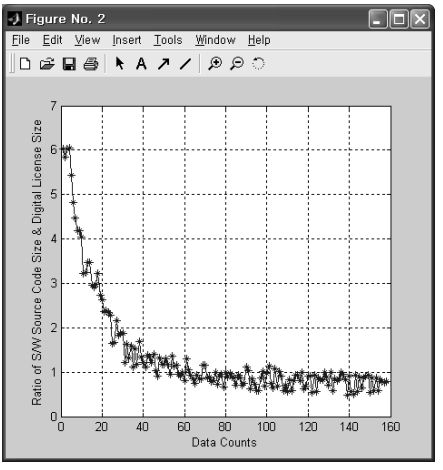


Fig. 7. Ratio of S/W Source Code Size & Digital License Size

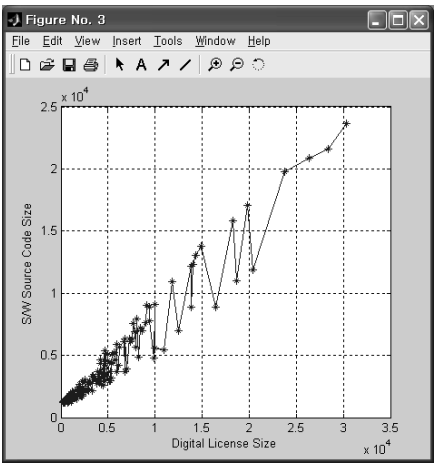


Fig. 8. Size of Digital License vs. S/W source cod

better 5 times than existing size of source code. Overhead is author information of S/W developer and company. According as size of SW source code increases, size of digital license changed monotone increasing with noise. We inferred noise from Digital License vs. SW source code size in fig. 8. Noise is produced by remark(comment) in Source code and coding style of SW developers.

5.2 Digital License Indexing and Searching

First, we search digital license instead of software source code in file system and create index of invert file using Lucene[16] of opensource java search engine. Fig. 9 shows result of Digital License Indexing. And then searcher retrieves information of digital license and position using created invert file. Fig. 10 shows result of Digital License Searching.

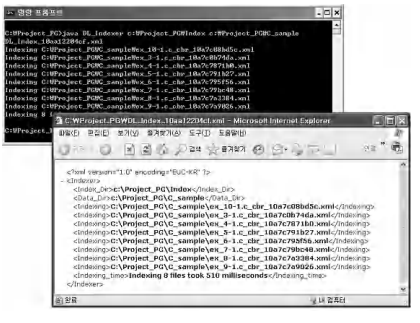


Fig. 9. Result of Digital License Indexing

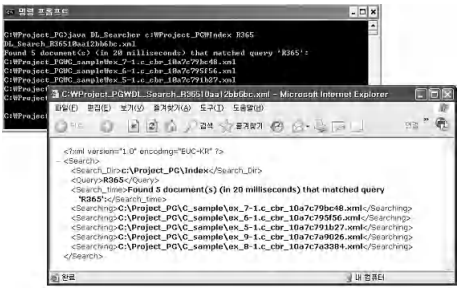


Fig. 10. Result of Digital License Searching

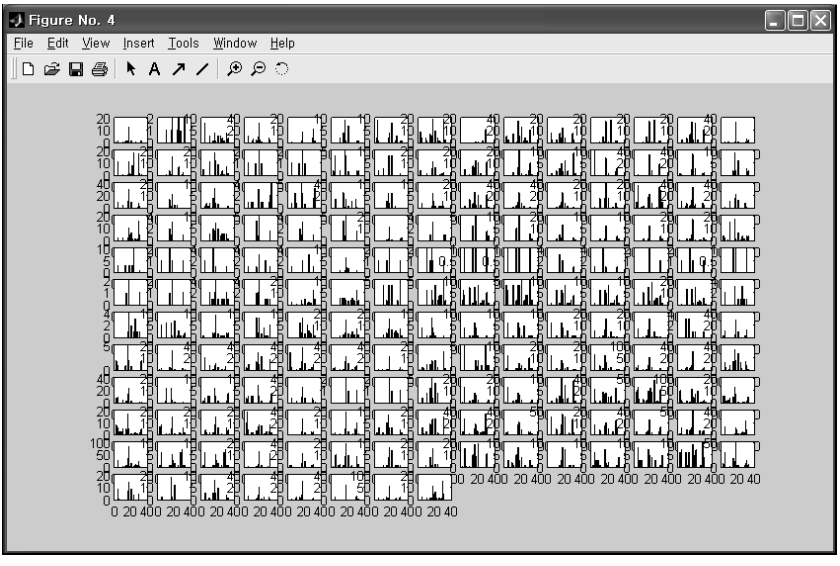


Fig. 11. Index Pattern Comparison by 32 reserved word of Search step 1

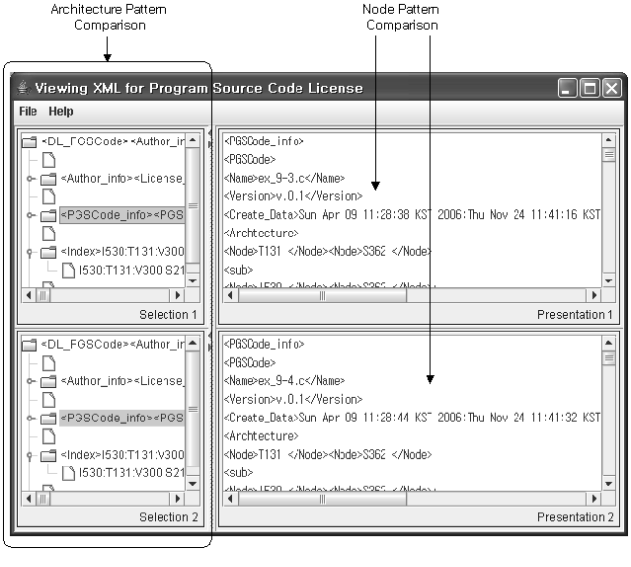


Fig. 12. Architecture & Node Pattern Comparison of Search step 2 & 3

Instead of SW source codes, We can compare digital license by index pattern as search step 1. Fig. 11 shows index pattern comparison by 32 reserved word using digital licenses at a look. And we can compare architecture & node informations

of two digital license using XML viewer as search step 2 & 3. Fig. 12 shows architecture pattern and node pattern of digital licenses using digital license viewer.

6 Conclusion

The IPR system plays a core role in the development of information society in the 21st century as it did in the birth and growth of previous industrial society. The IPR system and description for the extended software source code from digital contents are very significant to enhance international competitiveness. Currently other methods than the encryption do not exist in connection with the software source code, a resource factor of digital works, and the research for this is at early phase. The encrypted means are applied only to the first trade between the software source code developer and buyer, and they do not help to protect the copyright for the following trades. A problem happens if the original software source code is made public at dispute on its ownership. This study has suggested the design and business model of digital license of software source code to protect the copyright infringement and technique leakage caused by its opening. And we make a study of the indexing and searching to search digital license. The software source code is significantly long and complicated. It is supposed to be hard to find the uniqueness in the middle of inspecting the ownership if different softwares are operated by similar controlling structure, or if they are constituted of similar algorithms, or if one of them is a partially modified version of the other. There might be other cases to judge the uniqueness evidently. Therefore further study on the pattern to ensure the uniqueness is required. In addition, in-depth research is needed for more clear description of the copyright protection and information of software. Continuous study on the generation of various pattern information to integrate the structural programming language and object-oriented language and to include the information on software source code are required.

References

1. Iannella, R.: The Opend Digital Rights Language: XML for Digital Rights Management. Information Security Technical Report, vol. 9(3), pp. 1363–4127. Elsevier, Amsterdam (2004)
2. Neumann, P.G.: Principled assuredly trustworthy composable architectures, Technical report, Computer Science Laboratory, SRI International, Menlo Park, California, Final report, SRI Project 11459 (2003)
3. Chow, S., Eisen, P., Johnson, H., van Oorschot, P.C.: A White-Box DES Implementation for DRM Applications. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, Springer, Heidelberg (2003)
4. Jordan, C.N.C., van Buuren, R., Hartel, P.H., Kleinhuis, G.: Security Attributes based Digital Rights Management. In: Int. Workshop on Interactive Distributed Multimedia Systems/Protocols for Multimedia Systems (IDMS/PROMS), Springer, Heidelberg (2002)

5. Feigenbaum, J., Freedman, M., Sander, T., Shostack, A.: Privacy Engineering for Digital Rights Management Systems. In: Proceedings of the 2001 ACM Workshop on Security and Privacy in Digital Rights Management, pp.76–105 (2001)
6. Hartung, F., Ramme, F.: Digital rights management and watermarking of multimedia content for m-commerce applications. IEEE Communications Magazine 38, 78–84 (2000)
7. Rieger, M., Ducasse, S.: Visual Detection of Duplicated Code. In: Proceedings ECOOP Workshop on Experiences in Object-Oriented Re-Engineering (1988)
8. Yang, W.: Identifying Syntactic Differences Between Two Programs. Software-Practice and Experience 21(7), 739–755 (1991)
9. `http://www.few.vu.nl/dick/sim.html`
10. `http://glimpse.arizona.edu/javadup.html`
11. `http://www.ccsr.cam.ac.uk/mw263/YAP.html`
12. `http://www.cs.berkeley.edu/aikem/moss.html`
13. dong-uk, C.: S/W Program Plagiarism Inspection Techniques and Analysis of S/W Tools for Protection of Digital Properties. In: Proceeding of the Korea Contents Association. vol.1(1) (2003)
14. Garms, J., Somerfield, D.: Professional Java Security, Wrox (2001)
15. Lautenbach, B.: Introduction to XML Encryption and XML Signature. In: Information Security Technical Report, vol. 9(3), pp. 1363–4127. Elsevier, Amsterdam (2004)
16. Hatcher, E., Gospodnetic, O.: Lucene in Action, Manning (December 2004)
17. Cha, B.: Business Model and comparison of S/W Source Code vs. Digital License for IPRs, LNAI 4496, pp. 825–832 (June 2007)

Creating Numerically Efficient FDTD Simulations Using Generic C++ Programming

I. Valuev[1], A. Deinega[2], A. Knizhnik[2], and B. Potapkin[2]

[1] Joint Institute for High Temperatures of the Russian Academy of Sciences, Izhorskaya 13/19, Moscow, 125412, Russia
valuev@ihed.ras.ru

[2] KINTECH Kinetic Technologies, Kurchatov Sq. 1, Moscow, 123182, Russia
info@kintech.ru

Abstract. In the present work we propose a strategy for developing reusable multi-model simulation library for solving Finite-Difference Time-Domain (FDTD) problem for Maxwell's equations. The described EMTL (Electromagnetic Template Library) architecture is based on the selection of a small number of primitive low-level physical and numerical concepts which are used as parameters and building blocks for higher-level algorithms and structures. In the present work we demonstrate that a large set of FDTD techniques may be formulated using the same primitives. The basic concept for this representation is a discretized field contour entering the integral form of Maxwell's equations. We also describe the proposed architecture in terms of FDTD C++ template class library and discuss the performance and the usage of this library for various FDTD-based simulations.

1 Introduction

Interaction of electromagnetic fields with periodical materials is an important phenomenon in different device applications, such as radio antennas, wave converters, filters, etc. Recently a new field of application of periodical systems in visible optical wavelength range was proposed [1]. These systems are generally called as photonic crystals, that is, systems which are composed of a periodic arrangement of dielectric or metal material in two or three dimensions [2]. Computer modeling of such periodical systems is a power tool in design of new optoelectronic devices with the desired properties.

A general solution method of Maxwell's equations, which describe the evolution of electromagnetic waves in periodical materials, is the finite difference time domain (FDTD) method [3]. Since introduction in 70th years of the previous century this method became popular due to it certain advantages: (a) simplicity of explicit numerical scheme, (b) applicability to different geometries (only grid should be adapted), (c) explicit description of non-linear materials properties, (d) natural description of impulsive regimes. The inevitable disadvantages of this method evidently result from the above advantages: (a) large memory requirements for fine meshes; (b) strong relationship between time and space steps. The fine features of a considered system can require dense mesh for accurate description, which in turn, results in large memory requirement. For example, accurate

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 213–226, 2007.
© Springer-Verlag Berlin Heidelberg 2007

description of inverse opal structures with thin metal necks between voids may require fine mesh with cell sizes significantly smaller than characteristic field decay length in the metal. This results in large memory resources for fine mesh and makes corresponding calculations quite expensive.

Several approaches were proposed to overcome the disadvantages of the FDTD method [3]. In particular, using contour-path formulation [4] of Maxwell's equations it was possible to treat accurately sub-cell features without reducing FDTD cell sizes. The more general solution of efficient treatment of strongly non-homogeneous systems is to use non-uniform grids [5]. However, available widespread implementations of FDTD method (such as Meep [6], XFDTD [7], CFDTD [8]) have only limited support of these advanced FDTD methods.

In this work we present a general systematic implementation of FDTD method (called EMTL, Electromagnetic Template Library) based on the contour path formulation of Maxwell's equations, which can simultaneously describe sub-cell features and non-uniform grids. In particular, this implementation of FDTD supports both multi-block grid containers and multi level grids. Thus the total simulation grid can be represented as a set of different grids (which can in turn be non-uniform), covering efficiently each part of the simulated system. The interpolation and connection between different grids in this implementation is done using the contour path formalism. The multi-grid basis of this FDTD implementation provides natural parallelization strategy in the framework of domain decomposition method. Moreover, this implementation also provides full support of modern FDTD method simulation features, such as periodical and perfect matched layer (PML) boundary conditions, total field/scattered field subdivision of the simulation cell, extended library of optical materials properties etc.

The paper is organized as following: in the second section we describe the contour-path FDTD approach and geometry building blocks of EMTL; in the third section the mesh interplay and parallel strategies are presented; the fourth section gives brief idea of the EMTL implementation of the basic FDTD algorithms such as TF/SF, PML, etc.; the fifth section presents some recent benchmarks; conclusions are given in the sixth section.

2 Geometry Components of the Contour-Path FDTD

The integral form of charge-free Maxwell's equations ($c = \epsilon_0 = \mu_0 = 1$):

$$\frac{\partial}{\partial t}\int_S \boldsymbol{B}d\boldsymbol{S} = -\oint_l \boldsymbol{E}d\boldsymbol{l},\ \frac{\partial}{\partial t}\int_S \boldsymbol{D}d\boldsymbol{S} + \int_S \boldsymbol{J}d\boldsymbol{S} = \oint_l \boldsymbol{H}d\boldsymbol{l} \tag{1}$$

$$\oint_S \boldsymbol{D}d\boldsymbol{S} = 0,\ \oint_S \boldsymbol{B}d\boldsymbol{S} = 0 \tag{2}$$

can be used as a fundamental starting point for the formulation of FDTD discretization schemes. Indeed, following contour-path approach [4], let us consider a fixed contour in space, incircling flat polygon (Fig 1 a). For the second-order in space approximation, usually sufficient for linear materials, equations (1) may be rewritten in discretized central-difference form:

$$S(\frac{\partial}{\partial t}\boldsymbol{F}^c_{1,2} perm_{1,2} + \boldsymbol{J}^c_{1,2} - loss_{1,2}\boldsymbol{F}^c_{1,2}) = \pm \sum_i \boldsymbol{F}^l_{2,1}\boldsymbol{l}_i \tag{3}$$

The fields of different conjugated types $\boldsymbol{F}_{1,2} = \boldsymbol{E}, \boldsymbol{H}$ are separated: the fields $\boldsymbol{F}^c$ and source currents $\boldsymbol{J}^c$ measured at the center of the contour and multiplied by the polygon area S are at the left-hand side of (3), while the fields $\boldsymbol{F}^l$ contributing to curl and observed at the edge centers along the contour are entering the right-hand side multiplied by edge vectors $\boldsymbol{l}_i$. The permeability and loss ($perm_1 = \varepsilon$, $loss_1 = \sigma$, $perm_2 = \mu$) are related to the left (surface) side of the equation. They represent material properties:

$$\boldsymbol{D} = \varepsilon\boldsymbol{E}, \qquad \boldsymbol{B} = \mu\boldsymbol{H}, \qquad \boldsymbol{J}_1 = \sigma\boldsymbol{E} + \boldsymbol{J}_{source}, \qquad \boldsymbol{J}_2 = 0. \tag{4}$$

In order to make the explicit time-stepping possible, some kind of approximation, like semi-implicit approximation

$$\boldsymbol{F}(t) = \left[\boldsymbol{F}(t + 0.5\Delta t) + \boldsymbol{F}(t - 0.5\Delta t)\right]/2 \tag{5}$$

$$\boldsymbol{F}'(t) = \left[\boldsymbol{F}(t + 0.5\Delta t) - \boldsymbol{F}(t - 0.5\Delta t)\right]/\Delta t \tag{6}$$

is needed in the left-hand side of (3). Then (3) can be used to express the fields at the next time level via the fields at the previous levels:

$$\boldsymbol{F}^c_{1,2}(t + \frac{1}{2}\Delta t) = \alpha_{1,2}\boldsymbol{F}^c_{1,2}(t - \frac{1}{2}\Delta t) + \beta_{1,2}(\sum_i \boldsymbol{F}^l_{2,1}(t)\boldsymbol{l}_i - \boldsymbol{J}_{source}S), \tag{7}$$

where $\alpha_{1,2}$ and $\beta_{1,2}$ are material-dependent update coefficients for each type of the field. For approximations of greater orders in time or space, equations similar to (3) may be written containing more "reference" points at the polygon where fields have to be known to compose the surface and the curl part of the equation; or containing more "time layers".

If the contour passes the interface between different materials (Fig 1 a), an approximation of the material properties at the left-hand side is also required. The simplest approximation of that kind is taking the material property at the center of the contour (staircase model for Yee meshes). However, for some geometries, subcell averaging models [9] appear to be extremely useful (see Section 5.3).

The numerical data for fields at reference contour points is stored in computer memory which can be represented as many-dimensional vector $\mathbf{M}$. To bind the field value at any given space point $\boldsymbol{x}$ and memory locations (indices of $\mathbf{M}$) the following general interpolation expression is used in EMTL:

$$\boldsymbol{F}(\boldsymbol{x}) = \sum_{i \in I(\boldsymbol{x})} \mathbf{c}_i(\boldsymbol{x}) \cdot \hat{L}_i(\mathbf{M}), \tag{8}$$

where $I(\boldsymbol{x})$ gives (normally small) index set for a given point, $\mathbf{c}_i$ is interpolation coefficient. The "memory layout operator" $\hat{L}_i$ binds the interpolation term of index i with the memory array (Fig 1 b). Usually this operator is linear with respect to memory elements: $\hat{L}_i(\mathbf{M}) = \sum_k \alpha_{ik} M_k$. Different mesh layouts, interpolation

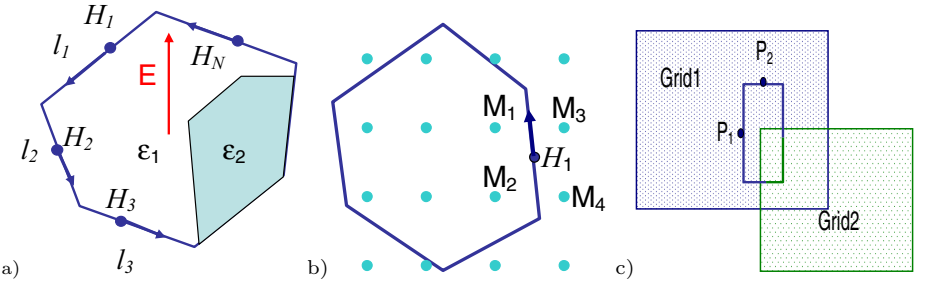


Fig. 1. a) — **H**-contour used to update **E** field in its center; b) — grid points interpolating the field along the contour; c) a contour crossing region boundaries

types and memory storage patterns may be fit into this expression. For example, for 3-linear interpolation in equidistant orthogonal 3D grid $N_0 \times N_1 \times N_2$, with space step Δg and per-component memory layout, the interpolation set consists of 24 elements:

$$I(\boldsymbol{x}) = \{i_{8k+4m_0+2m_1+m_2}\}_{k=0,1,2;\, m_0,m_1,m_2=0,1} \tag{9}$$

$$i_{8k+4m_0+2m_1+m_2} = N_0N_1N_2k + N_1N_2(n_0+m_0) + N_2(n_1+m_1) + n_2 + m_2$$

$$\boldsymbol{c}_i(\boldsymbol{x}) = \boldsymbol{c}_{8k+4m_0+2m_1+m_2} = \boldsymbol{e}_k \prod_{j=0}^{2}((\frac{x_j}{\Delta g} - n_j)(-1)^{m_j} + m_j), \quad \hat{L}_i(\mathbf{M}) = M_i,$$

where $n_j = x_j \bmod \Delta g$ are basic grid indices for the position $\boldsymbol{x}$, the increment of memory pointer by one corresponds to the change in z-coordinate of the grid (array index $j = 2$ changes first in memory). As it will be discussed later (Section 5.1), memory layout may have significant influence on the code performance.

Note that (8) is used both for collecting the input field values from memory before each field update according to (7) and for placing the (different) updated output fields into memory. Using the linearity of (8) and (3) with respect to $\hat{L}_i(\mathbf{M})$, each ith component, corresponding to the memory location $\hat{L}_i(\mathbf{M})$, may be updated separately. This allowed us to implement the interpolation expression (8) in EMTL in most general way. However, special mesh layouts may simplify (8) greatly, reducing the number of the input/output field update combinations. For example, in the standard Yee mesh construction [10], the index set $I(\boldsymbol{x})$ is reduced to one component for each update location $\boldsymbol{x}$ in the center of the contours and their edges, thus resulting in one update combination per contour with four input memory elements (edge centers) and one output element (contour center).

3 Design of the FDTD Simulation in EMTL

3.1 Simulation Stages and Basic Components

As any FDTD simulation, the EMTL-based application has three main computation stages: (1) geometry setup and pre-processing of update coefficients;

(2) main loop with iterative time propagation and recording of output data; (3) processing of output data. Before describing the first two stages of the simulation in more detail, we list briefly the key components of EMTL. The building blocks of EMTL are components that are used as template parameters in higher-level structures. These building concepts are STL-like iterators, points (vectors), contours, bodies, interpolation forms, mesh blocks. The methods of these components are used from the inner loops at geometry setup or time propagation stages and implemented as inline functions for performance reasons.

The basic *geometry* components in EMTL are points in 3D, contours (flat convex polygons) and continuous regions (bodies) in 3D. All bodies implement a concept of `SpaceRegion`, requiring the corresponding C++ classes to define the following functions, necessary for contour path analysis:

- testing whether a point (3D vector) is inside the region;
- for a given flat contour, return the area of a subcontour which is inside the region;
- for a line segment connecting given inside and outside points, return the intersection point of the body surface with the segment and the surface normal vector at the intersection.

The last function is used in subcell models where the interface direction between two materials is important [9]. All regions, except for the main bounding region, may be infinite in one or more dimensions, for example a `Polyhedron` can contain only one plane selecting a half-space with some material properties. Simulation in one or two dimensions is possible by reducing the size of non-existent dimensions to two contours and introducing periodic boundary conditions in the reduced directions.

The meshes, or `emMeshBlock` objects, are the building blocks of the FDTD model and define a set of update contours, memory layout and interpolation. Thus, EMTL separates the mesh design task (which is itself a complicated problem) from the FDTD simulations which use a particular mesh. Practically, any FDTD method available in EMTL may be used with arbitrary meshes and even their combinations. The `emMeshBlock`-conforming classes must implement the following key functionality:

- define a "control region" where the mesh can provide an interpolation of fields;
- return an interpolation for a given field type ($\boldsymbol{E}$ or $\boldsymbol{H}$ at a given location $\boldsymbol{x}$ in the form (8), binding memory array indices and coefficients;
- define an iterator of all $\boldsymbol{E}$ and $\boldsymbol{H}$ contours composing the mesh;
- provide optimized basic bulk update function and accept basic update coefficients α, β (7) for contours
- optionally accept indications to "unused" (not updated) contours to save memory.

Various kinds of meshes can be designed to conform to these model, including non-uniform, unstructured and non-orthogonal ones. Note that the chain of contours defined by the mesh must satisfy the second pair of the Maxwell's equations (2).

3.2 Geometry Analysis Stage

The contour collection that is returned by `emMeshBlock` is analyzed at geometry setup stage when the update coefficients for each contour are calculated. The analysis is performed only once before the iterative update stage, therefore this part of a simulation is less performance critical and may be generalized. The contour analysis algorithm is the main algorithm of EMTL, it is implemented in the most general way and does not depend on the mesh construction details. The only requirements for the mesh is the conformal definition of a contour: (a) contour is a sequence of points; (b) contour is flat and convex; (c) here is a binding between contour characteristic points (center for output and edge centers for input) and memory locations provided by `emMeshBlock`; (d) contour is connected with some "control" volume. The last contour property is used in smoothing models only where volume averaging is required. The contour analysis algorithm can be easily extended to include new types of equations, materials and subcell smoothing models.

To construct the whole task geometry, the components are assembled in the object of type `emBlockContainer<mesh_type1,mesh_type2>`. This is the main "access object" of the model, it can contain arbitrary number of mesh blocks, each of them belonging to one of the two mesh types. The `emBlockContainer<>` can itself be treated as an aggregate `emMeshBlock`, allowing to combine arbitrary number of mesh types in one model. The top level container must have a bounding region for which boundary conditions are defined. The boundary conditions determine the behavior when a certain contour crosses the container boundary and some of its input points are outside. In case of periodic boundaries, these points are reverted to the other side of the container using "transfer fix", while for reflective conditions the input fields located outside are assumed to be zero.

Multiple `emMeshBlock` objects may be placed in the container having control regions with different interpolation levels assigned to them. In case the meshes intersect, the actual field update is performed in the mesh of higher level, while the lower level meshes use interpolated values for the input points of their contours (Fig 1, c). The same mechanism works for partially intersecting meshes or when transfer between different CPU domains is needed. In the later case, the MPI transfer is taking place along with data interpolation.

3.3 Separation of Duties: Basic and Specialized Updates

The equation (7) with constant material properties (4) is the basic field update relation in EMTL. Any mesh (the object of class `emMeshBlock`) must be capable of performing this type of update on a specified range of mesh contours. This repeatedly executed basic bulk update takes usually the major part of CPU time during the simulation. Although a generic EMTL update step implementation based on (7) and (8) may be utilized, the usage of specialized algorithms taking memory layout for a particular mesh into account can greatly improve the code performance. Because of its relatively simple localized pattern, the basic update may be decomposed into the processing of independent memory streams. The

minimal number of streams (four for input and one for output in homogenous medium case) is achieved in the Yee mesh layout. The advance of the streams (read or write) with unit memory increment is interchanged with stream rebinding when contour iterator reaches mesh boundaries (see Section 5.1).

Possible modifications to (3), resulting in applying different update equations to some contours are introduced in the form of corrections to (7) or *Fixes* in terms of EMTL. The modified equations are usually needed to include various features in the FDTD model, such as source field generation, absorbing boundary conditions, dispersive materials, etc. Note that unlike the traditional approach of including the specialized updates into the main update loop and testing the equation type flags on each iteration, these updates are performed as separate *Fix loops* in EMTL. The identifiers of the contours requiring special processing are recorded in the packed fix lists. There are fixes that are performed before (changing the input fields) and after (correcting the output fields) the basic field update of all contours. For example, the data transfer between different meshes in the area of mesh intersection is implemented as a pre-step fix. The transfers between different CPU domains in case of MPI parallel execution require both pre-step fix (initializing the boundary sends) and post-step fix (completing receives and updating the boundary-dependent contours). Unlike the basic update, the fixes contain model-specific algorithms and are implemented in a general mesh-independent way. However, when the number of contours affected by a certain fix is large, the mesh-specialized implementation may be useful to gain performance. EMTL provides a means for the code developer designing a mesh to define such specialized updates.

As for all explicit time integration schemes, the maximal time step dt for field update in FDTD is connected with the space resolution dx and propagation speed c by the Courant condition: $dt \approx dx/c$. Thus, when meshes with different space resolutions coexist in the model, it is necessary to propagate them with different time steps. In EMTL, the meshes are advanced in the order of their time steps, the mesh with the largest possible time step is updated first determining the next target time. All other meshes are advanced in order till they reach the target time. Special "synchronization fix" is used for transfers between meshes with different time layers. For that transfers, the field data from the previous time level is also required to interpolate the input fields both in time and space. This pre-step fix stores the required for interpolation to other meshes field values of current time level, before they are overwritten by the next field update.

3.4 Optimizing Memory Usage

The fix loops iterate over a subset of the contour set provided by the mesh. The sequence of "fixed" contours depends mainly on large-scale geometric setup of the model and is in most shows regular patterns. Moreover, the data utilized by contour fixes often contains the same corrections for different contours. To account for this regularity, a simple data packing techniques are used in EMTL. The packing is based on the detection of linear trend subsequences (identical increment) in the sequences of integers. The subsequences are then encoded by

replacing them with an indicator code (-1) followed by the (increment, number) pair. Although very primitive, this compression permits to save much memory for bulk corrections and does not induce significant overheads when unpacking. Both the iterator sequences and data index sequences (pointers to identical data values) can be effectively packed using this strategy. For complex correctors such as UPML fix, utilizing large amount of data, the fix update sequence reordering can be utilized to improve cache efficiency when accessing the packed data. In this case, the order of contour iterations may be adjusted corresponding to the order of unique data values stored in memory.

3.5 Parallelization Strategy

EMTL supports parallelization using space domain decomposition which is a common technique for time domain solvers. The CPU domains may be arbitrary `SpaceRegion`s, subdividing the main simulation region. The EMTL contour analysis algorithm detects contours crossing domain boundaries and registers "transfer fixes" for the contour input points belonging to non-local meshes or CPU domains. The data interpolation coefficients required to represent data from a source different from the mesh the contour originates from (different domain or different mesh) are recorded in transfer fix. They are convolved with the data taken from the remote source on each field update iteration. The only difference for parallel execution is that the recorded data transfer requires MPI message rather than direct memory copy.

The automated balanced decomposition based on bisection algorithm is implemented in EMTL and may be utilized to decompose the main region into box-shaped domains according to the number of processors detected at program start. To split the region into balanced domains, the algorithm uses a number of recursive region bisections by planes in axial directions. The bisections are performed in such a way that two resulting subregions are balanced in terms of $n_i \int_{V_i} w(\boldsymbol{x}) dV$, where n_i is the number of processors assigned to subregion i, w and V_i are "workload density" and the subregion volume correspondingly. The workload density represents the cost of computations for a given space location. It can be approximated by roughly sampling the fraction of different materials/equations along the bisection axis. The same contour path algorithm, determining the crossing fraction of a contour with a body is used for this purpose. The correctors are assigned experimentally determined workload weights (the basic bulk update has the weight of 1.). For example, for UPML absorbing boundary condition fix (see Section 4.3) the measured weight is approximately 2.5. The bisection algorithm is then applied to the subdivisions and stops recursion when the number of assigned processors for a subregion reaches one.

Techniques trying to optimize cache efficiency of the basic bulk update by reducing the number of memory stream rebindings are possible within EMTL decomposition algorithm. For example, the authors of [11] argued that the "pencil" memory layout can save up to 60 per cent of computational time on Beowulf type architectures. In this technique the space dimension corresponding to the most rapidly changing data array index is kept as extensive as possible, while

domain decomposition is performed in the remaining dimensions. The EMTL bisection algorithm permits to specify the number of bisections (including zero) along selected axis explicitly, and the effects of pencils may be studied directly (see Section 5.1).

4 Implementing Basic FDTD Algorithms as Correctors

In this section we briefly describe the FDTD techniques available in EMTL in the form of "fixes", or corrections to the basic bulk update equations (7).

Wave Generation by the Total Field/Scattered Field Method. The TF/SF technique [3] is a very effective source field generation method when the simulation domain may be split into the Total Field (TF) and Scattered Field (SF) regions. It is assumed, that the incident wave propagation is known analytically for the case there are no scattering objects. The update equations in the SF zone are solved for the *scattered field* which is a difference between the total field and the known unscattered incident field. The TF zone contains scattering objects and the update equations for the total field are solved there. The update equations have exactly the same form (7) in both zones, except for the places where the update contour passes the TF/SF boundary. The output point and the input points of such a contour may be classified to be TF or SF points according to their positions. If the update involves input points of the type different from the output, then a corresponding correction (incident field at the contour edge) is added or subtracted:

$$\boldsymbol{F}^c(t+\frac{1}{2}\Delta t) = \boldsymbol{F}_{bulk} \pm \beta(\sum_{diff.\ zone} \boldsymbol{F}^l_{inc}(t)\boldsymbol{l}_i), \tag{10}$$

where $\boldsymbol{F}_{bulk}$ is the basic update at the right hand of (7), "+" sign is used when the output point belongs to the TF zone and "−" otherwise. The TF/SF fix records the update coefficients $\pm\beta\boldsymbol{l}_i$ for each combination of input/output memory indices composing the interpolations of $\boldsymbol{F}$. These indices are also classified to be of TF or SF type.

Absorbing Boundary Conditions. This technique prevents reflection of the wave from the container boundaries thus simulating a finite object (container) in an infinite surrounding medium. We implemented the Uniaxial Perfectly Matched Layer [3] (UPML) technique as a fix within EMTL. To guarantee absorbtion of a signal on some interface at any incidence angle, the "matched" frequency-dependent dielectric/magentic tensors are introduced in absorbtion zone instead of ε and μ in (4):

$$\boldsymbol{D} = \varepsilon\, s(\omega)\, \boldsymbol{E}, \qquad \boldsymbol{B} = \mu\, s(\omega)\, \boldsymbol{H}, \tag{11}$$

where $s(\omega)$ is a diagonal tensor. The frequency dependence of its diagonal elements is a rational function of the form $(a_1 + jb_1\omega)/(a_2 + jb_2\omega)$, which leads to

the differential equation of the first order linking the components of the contour-centered fields $\boldsymbol{D}^c$ and $\boldsymbol{E}^c$ in the time domain, for example:

$$k_1 \frac{\partial}{\partial t} D_x^c + k_2 \frac{\partial}{\partial t} E_x^c + k_3 D_x^c + k_4 E_x^c = 0. \tag{12}$$

This equation supplements the basic update (7), which in this case is formally performed for $\boldsymbol{D}$. The equations for $\boldsymbol{B}^c$, $\boldsymbol{H}^c$ pair have the similar form. The function of the EMTL fix for the contours entering UPML region is to reserve a space in memory for extra independent variables ($\boldsymbol{D}$, $\boldsymbol{B}$) and coefficients k_i, and to solve (12) at each update iteration using explicit time stepping.

Dispersive Materials. The addition of the auxiliary update equations to the contour basic update is also used when frequency-dependent dispersion is introduced in the dielectric function. This representation is usually originating from the fits of the dielectric function of real materials in frequency domain, approximating it with multiple terms of the Debye, Drude or Lorentz form [12]. These terms may be generalized by the formula [3]:

$$\varepsilon(\omega) = \sum_i \frac{a_i + j b_i \omega}{c_i + j d_i \omega + e_i \omega^2}, \tag{13}$$

where a_i ... e_i are real coefficients. Each term of (13) is associated with a partial polarization $\boldsymbol{P}_i = \varepsilon_i(\omega)\boldsymbol{E}$. The equations for $\boldsymbol{P}_i$ are converted to the time domain, resulting in the differential equations of the order determined by the highest power of ω in the denominators of $\varepsilon_i(\omega)$. The EMTL fix managing the variables $\boldsymbol{P}_i$ and their update equations for the contours crossing dispersive material regions is organized in the way similar to the UPML fix described in the previous subsection.

5 Illustrative Examples and Benchmarks

In this section we present the actual simulation results and comparisons obtained so far with EMTL-based applications. In the benchmark subsection we compare EMTL with GPL-licensed MEEP code developed in MIT [6]. Although not implementing non-Yee meshes and multiple meshes, this parallel code offers a variety of FDTD techniques and can be used as a library tool.

5.1 Mesh Memory Layout Testing

Mesh memory update is the central procedure in the time domain equation solvers. The mesh data itself is related to physical properties measured at certain locations in space. This localization is often used as a basis for memory layout, where the data entries responsible for physical properties characterizing a given mesh cell are stored in adjacent memory locations. Usually the equations imply update loops of multiple kinds, when only a limited number of physical properties

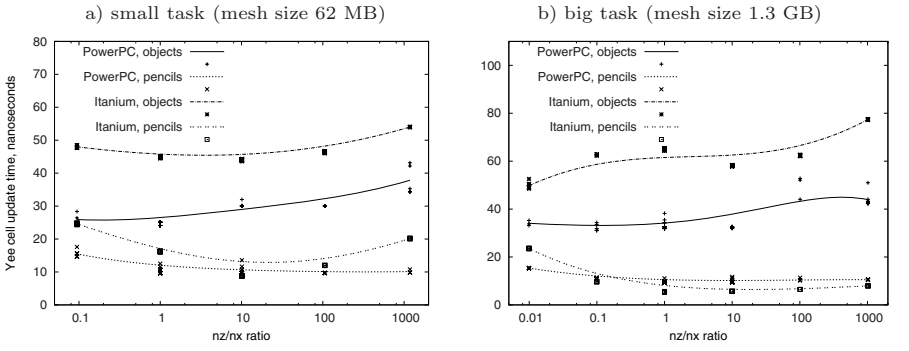


Fig. 2. Effects of the memory layout and simulation domain proportions ($nx = ny$ by varying nz) on FDTD performance in vacuum update (constant material properties). The code was compiled on PowerPC with IBM xlc 7.0 (-q64 -O5) compiler and on Itanium with Intel icc 9.0 (-O3) compiler. The tests were performed at the Joint Supercomputer Center of RAS [13].

are affected by the individual update loop, applied for all cells. In this case the "cell as an object" memory organization may reduce cache efficiency because it increases the memory pointer stride when accessing mesh data. An alternative is the memory layout where mesh data is grouped by the physical property in possibly long and contiguous arrays, or "pencils" [11].

Being a template library, EMTL supports arbitrary memory layout strategies and allowed us to study the memory layout effects for FDTD (Fig. 2). Two orthogonal Yee meshes of size ($nx \times ny \times nz$) were used for testing: the one where six field components per each Yee cell are stored in adjacent memory ("objects"), and the other where each component type is stored in a separate array ("pencils"). In both cases the most rapidly changing spatial mesh index was the one along z-direction of the simulation domain. Choosing different nz by keeping $nx = ny$ and constant domain volume, we measured the update time per Yee cell. On both considered architectures (Itanium 2 and PowerPC 970) the "pencil" layout resulted in much higher performance. The memory stream length (nz/nx) parameter also affects performance, but to less extent than the layout itself.

5.2 Parallel Benchmarks

To study the parallelization efficiency of our code, we performed a set of scaled and unscaled tests on different platforms (Fig. 3). For the *unscaled* tests the task size is kept constant and the execution time $t(N)$ is measured while varying the number of processors N, the parallel efficiency is calculated as $t(1)/Nt(N)$. For the *scaled* tests the mesh size is proportional to the number of processors and the parallel efficiency is computed as $t(1)/t(N)$.

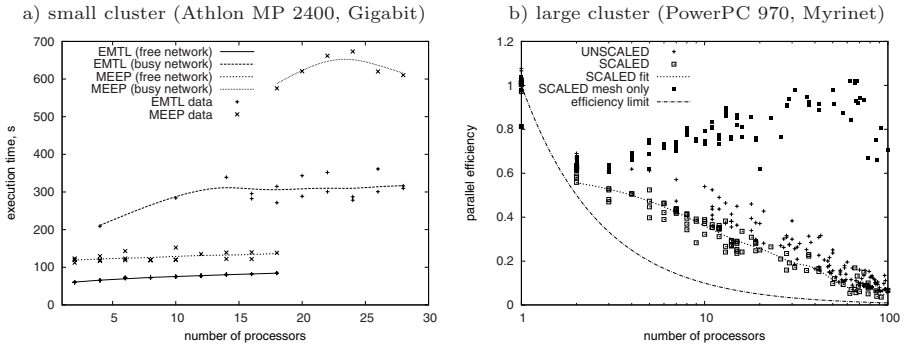


Fig. 3. Parallel benchmarks. a) — scaled test compared to MEEP [6] code on a small cluster [14], the basic task size is 140^3 cells vacuum box, scaled in one dimension proportionally to the number of processors, 100 timesteps; b) — tests on a large cluster [13] in its regular multi-user production regime, basic task size is 300×100^2, limiting minimal efficiency curve $1/N$ is drawn for comparison. The efficiency of memory update (not taking network transfer times into account) is shown by filled squares.

The execution time trends for a small cluster, entirely dedicated to the task, may be branched according to the two regimes of cluster network load (Fig. 3, a), the interconnecting Gigabit switch presumably being the critical device limiting performance. The difference in performance between MEEP and EMTL growth in inverse proportion to the bandwidth of the switch, thus it can be mainly attributed to the parallelization. The parallel efficiency tests of the typical 1000-timesteps task on the large production system (Fig. 3, b) also show strong sensitivity of the code performance to the network state of the whole system. Large variation of the results, while *unscaled* efficiency is surprisingly higher on average than the scaled one, contributes to this conclusion. For the particular cluster studied, using medium number of processors (10–40) rather than small number (2–10) in parallel execution is still profitable (relative efficiency factor is about 0.7).

5.3 Testing Subcell Averaging Methods

The contour architecture of EMTL simplifies implementation of various subpixel averaging methods, used to improve FDTD accuracy when modeling discontinuous materials or small objects (Figs. 4, 5). We performed a series of tests using a) contour/volume averaging of material properties for metals; b) contour/volume averaging of ε or ε^{-1} for dielectrics; c) mixed tensor averaging [9]. In the latter case the ε for the contour is replaced by the dielectric tensor of the form $\epsilon^{-1} = P < \varepsilon^{-1} > +(1 - P) < \varepsilon >^{-1}$, where $P_{ij} = n_i n_j$ is the projection matrix onto the normal to material interface. We tested currently the mixed averaging using only the diagonal part of P (referred to as `dTensor` in the Fig 4). A gaussian pulse with characteristic width of 20 Yee space steps, impinging a single metallic or dielectric sphere with radius of 0.5–7 space steps, was used for testing. Scattering cross section from the sphere obtained for different subcell averaging methods was compared with the Mie

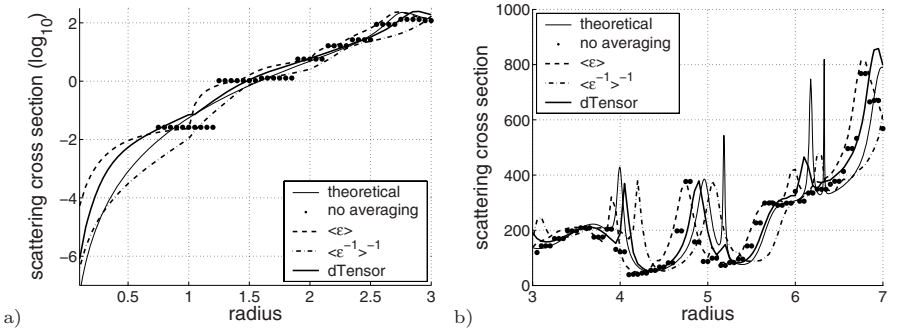


Fig. 4. Scattering from a small dielectric sphere ($\varepsilon = 12$) at $\lambda = 20$ using various subcell dielectric function smoothing methods compared to the Mie theory. The averaging is performed over "contour surface", except for dTensor case, where control volume average is used. The unit length is the Yee space step.

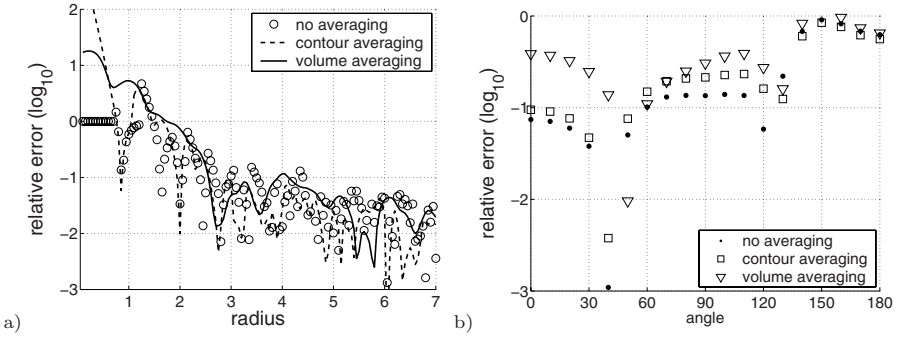


Fig. 5. Scattering from a small metallic sphere compared to the Mie theory, the metal is modeled by two Lorentz [3] terms $\delta\varepsilon_1 = \delta\varepsilon_2 = 0.5$, $\delta_1 = \delta_2 = 2\pi\times0.001$, $\omega_{01} = 2\pi\times0.04$, $\omega_{02} = 2\pi \times 0.06$. a) — relative error of the scattering cross section at ω_{02} as a function of the sphere size; b)— relative error of S_{11} scattering matrix element of $R = 5$ sphere at ω_{02} as a function of the scattering angle. The unit length is the Yee space step.

theory. For the metallic sphere, the scattering matrix element S_{11} [15] at resonant frequency was also studied as a function of observation angle (Fig. 5, b). To obtain fields at large distances and different scattering angles, Near-to-Far field transformation [16] was used in transforming the fields measured at 2.5 radius distances from the sphere to the far fields.

The results of the tests show, that averaging is important for small objects. The direct $< \varepsilon >$ and harmonic $< \varepsilon^{-1} >^{-1}$ averages lead to comparable but opposite in sign error, the proper mix of both [9], even using diagonal tensor only, leads to much smaller errors. The averaging over a "control volume" of a contour rather than over a contour "encircled surface" may be important for properly distinguishing very small objects but does not lead to any apparent improvement in the accuracy.

6 Conclusions

In the present work we demonstrated that a large set of FDTD techniques may be formulated using the same primitives. The basic concept for this representation is a discretized field contour entering the integral form of Maxwell's equations. The main aim of this paper was to describe contour-based mesh architecture utilized in EMTL and demonstrate its universality and performance. Some benchmarks and test obtained so far with EMTL are presented, the others are subject to the future work.

The library itself is a research tool oriented on the model code developers who need to use FDTD as part of their simulations. Being a library, rather than a standalone application, EMTL provides a set of template classes to implement a variety of modern FDTD techniques in the efficient parallel C++ code.

Acknowledgements. This work is partially supported by the research program #14 of the Russian Academy of Sciences. The benchmarks were performed using hardware of the Joint Supercomputer Center of RAS and the Department of Molecular and Biological Physics of the Moscow Institute of Physics and Technology.

References

1. Joannopoulos, J.D., Meade, R.D., Winn, J.N.: Photonic crystals: molding the flow of light. Princeton University Press, Princeton (1995)
2. Johnson, S.G., Joannopoulos, J.D.: Acta Materialia 51, 582 (2003)
3. Taflove, A., Hagness, S.H.: Computational Electrodynamics: The Finite Difference Time-Domain Method. Artech House, Boston (2000)
4. Jurgens, T.G., Taflove, A., Umashankar, K.R., Moore, T.G.: IEEE Trans. Antennas and Propagation 40, 357 (1992)
5. Dey, S., Mittra, R.: IEEE Microwave and Guided Wave Lett. 7, 273 (1997)
6. MIT Electromagnetic Equation Propagation, `http://ab-initio.mit.edu/wiki/index.php/Meep`
7. Remcom's full wave FDTD solver, `http://www.remcom.com/xfdtd6/index.htm`
8. RM Associates' conformal FDTD code, `http://www.rm-associates.biz/software.html`
9. Farjadpour, A., Roundy, D., Rodriguez, A., et al.: Optics Letters. 31, 2972 (2006)
10. Yee, K.S.: IEEE Trans. Antennas and Propagation 14, 302 (1966)
11. Dobbler, W., Haugen, N.E.L., Yousef, T.A., Brandenburg, A.: Phys Rev E 68 (2003) 026304, `http://www.nordita.dk/software/pencil-code`
12. Roberts, S.: Phys Rev. 114, 104 (1959)
13. Joint Supercomputer Center of the Russian Academy of Scienece, `http://www.jscc.ru`
14. Computational cluster of the Department of Molecular and Biological Physics, MIPT `http://biolab1.mipt.ru`
15. Bohren, C.F., Huffman, D.R.: Absorption and Scattering of Light by Small Particles. Wiley-Interscience, New York (1983)
16. Umashankar, K.R., Taflove, A.: IEEE Trans. Electromagnetic Compatibility 24, 397 (1982)

Mutual Authentication Protocol for RFID Tags Based on Synchronized Secret Information with Monitor

Song Han, Vidyasagar Potdar, and Elizabeth Chang

DEBII Institute
Curtin University of Technology
GPO Box U1987, WA
s.han@curtin.edu.au

Abstract. RFID, as an anti-conterfeiting technology, is pushing huge potential industrial, medical, business and social applications. RFID-based identification is an example of emerging technology which requires authentication. In this paper, we will propose a new mutual authentication protocol for RFID tags. The RFID reader and tag will carry out the authentication based on their synchronized secret information. The synchronized secret information will be monitored by a component of the database server. Our protocol also supports the low-cost non-volatile memory of RFID tags. This is desirable since non-volatile memory is an expensive unit in RFID tags.

1 Introduction

Radio frequency identification (RFID) is a method of remotely storing and retrieving data using small devices called RFID tags. RFID system could lower the cost of tagged items in the process of identification [9], [16]. Therefore, RFID technology is expected to diffuse into our everyday life. In a server-based scenario, an RFID system is composed of three players: the database server, the RFID tag(s), and the RFID reader(s).

Tag: is comprised of an IC chip and antenna, and sends information to the RFID reader in response to a wireless probe.

Reader: is a device that transmits a radio frequency probe signal to RFID tag, receives the information sent by tag, and sends the information to the database server.

Database Server: is a secure server that has a database and manages various types of information related to each tag, e.g., ID, reader location, read time, and temperature of sensor. The server resolves the ID of tag from the information sent by tag through authenticated RFIDreader.

The RFID reader makes an access request to the RFID tag and then returns the reply from the RFID tag to the database server. The identification and authentication are carried out under the help of the database server. After the RFID tag and reader authenticate each other, the database server returns the information of the RFID tag to the reader. Generally, most works assume the communication channel between the database server and the RFID reader is secure.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, pp. 227 – 238, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In this paper, we will design an RFID authentication protocol for RFID tags and readers. This protocol is based on synchronized secret information shared between the database server and the RFID tag. Our method is based on the framework of the previous works [1], [2], [5], [7], [11], [12], and [14].

1.1 RFID Security and Privacy Issues

The central issue the RFID technology raised is the privacy issue [7], [8], [10], [13], [15] as well as the cost [3], [4], [12]. In this paper we will focus on the privacy issue and RFID tag authentication. In fact, the mass deployment and acceptance of RFID technology is nowadays mainly limited by privacy concerns [9]. Products labeled with RFID tags reveal sensitive information when queried by readers, and they do it indiscriminately. This may induce the violation of location privacy, i.e. tracking. Two security requirements related to privacy are:

Indistinguishability: Tag output must be indistinguishable from truly random values. Moreover, they should be unlinkable to ID of the tag. If the adversary can distinguish that a particular output is from a target tag, he can trace the tag.

Forward security: Even if the adversary acquires the secret tag data stored in the tag, he cannot trace the data back through past events in which the tag was involved. Needless to say, the adversary who only eavesdrops on the tag output, cannot associate the current output with past output.

Besides the privacy and tracking issues, there are some other worth mentioning: impersonating, spoofing, eavesdropping, traffic analysis, etc.. In order to address the cost issue, our paper will design an authentication protocol which can reduce the cost of RFID tags through trimming down the non-volatile memory. This is similar to the protocols in [11] and [14].

1.2 Summary of the Attributes of the Proposed Protocol

In this paper, we will propose a new mutual authentication protocol for RFID tags. Our method is based on one-way hash function. Our protocol has the following attributes:

- The protocol is based on synchronized secret information.
- The synchronized secret information is monitored by an internal configuration of the database server. Thus, desynchronization attack can be detected and diminished.
- Hash chain requirement is not necessary in our protocol.
- The memory of the RFID tag is portioned into two parts: volatile memory and non-volatile memory.
- The structure of the tag memory can reduce the cost of RFID tags.
- The protocol supports user privacy protection.
- The protocol supports forward security and untracking property.

1.3 Organization of the Rest of the Paper

The reminder of our paper is organized as follows: In Section 2, some related works will be introduced. In Section 3, some notations will be first presented; then some

basic assumptions related to RFID tags and readers as well as database server will be provided; following that, the mutual authentication protocol based on synchronized secret information with monitor will be proposed; thereafter, the structure of the database server will be explained. In Section 4, the security analysis and comparison will be provided. In Section 5, the efficiency and its comparison will be investigated. Finally, we will conclude our paper.

2 Related Works

The authentication protocol we will propose in this paper is a hash-based authentication method. Therefore, the related works introduced here focus on the hash-based authentication protocols for RFID tags.

Ohkubo et al. [2] proposed a hash-based authentication protocol. The protocol changes RFID identities on each read based on hash chains. Their method also requires a hash function on the RFID tag, but does not require a random number generator. However, the protocol in [2] is vulnerable to certain re-play attacks.

The proposed modification in [1] address the replay attack problem but does not consider the issue of availability, and their protocol is vulnerable to attacks where the attacker forces an honest tag to fall out of synchronization with the server so that it can no longer authenticate itself successfully.

The scheme by Jules [3] only provides security against "fly-by" attacks where the attacker is allowed to interact with the tag for a fixed time budget but does not provide protection in the case of tag capture.

Molnar et al. [7] proposed a hash-tree based authentication protocol for RFID tags. They exposed privacy issues related to RFID in libraries, described current deployments, and suggested novel architectures for library RFID. However, the amount of computation required per tag is not constant, but logarithmic with the number of tags in the hash-tree. Also, if a tag is lost, anonymity for the rest of the hash-tree group may be compromised. Finally, the protocol does not provide for forward-anonymity.

Another hash-based authentication protocol was introduced by Henrici et al. [5]. Their solution does not provide full privacy guarantees, in particular, the tag is vulnerable to tracing when the attacker interrupts the authentication protocol mid-way.

Lee et al. [11] proposed an RFID authentication scheme which uses a hash function and and synchronized secret information. However, the RFID tag needs a PRNG. Moreover, the server does not know how many times an RFID tag may have not yet updated its secret information. Thus, desynchronization attacks may be conducted by adversaries.

Dimitriou [12] also proposed an anonymous RFID protocol against tag cloing. Their protocol is based on the use of a secret shared between tag and database that is refreshed to avoid tag tracing. However, their protocol is vulnerable to desynchronization attacks against availability.

Lee et al. [14] proposed a protocol to solve the desynchronization problem by maintaining a previous ID in the database server. However, an adversary who queries T actively without updating ID can trace the RFID tag because the hashed ID is continually identical. Moreover, the adversary can trace the previous event of tag because

ID is updated b XORing the previous ID with a random number emitted through radio frequency (RF).

3 Mutual Authentication Protocol with Monitor

In this section, we will develop the mutual authentication protocol for RFID tags based on synchronized secret information. Some notations will be first introduced and then used throughout the rest of our paper. Next, we will give the basic assumptions that related to what capability the RFID tags and readers as well as the database server have. Thirdly, we will present the new authentication protocol for RFID tags. Following that, we will explain the structure of the database of the server in detail. This is important since some characteristics of the structure can detect some adversaries that may compromise or prevent RFID tags from updating their secret elements. This point will be demonstrated in the Section 4.

3.1 Notations Used in Our Protocol

The following table provides the notations used in the proposed authentication protocol.

Table 1. Notations used in the proposed RFID tag and reader authentication protocol

Symbol	Representation
h()	One hash function available to all parties
T	An RFID tag
S	RFID system database server
R	An RFID reader
i, j	Non-negative integers
M	Monitor for checking RFID tag whether updates its secret element after a successful authentication
ID	RFID identification number
PRNG	A pseudorandom number generator
β	The bit-length of the output of h()
$\oplus$	Exclusive-OR function (XOR)
θ	A threshold for M
α	A seed for PRNG
VM	Volatile memory of RFID tag
NVM	Non-volatile memory of RFID tag

3.2 Basic Assumptions

The RFID tag T has a one-way hash function h(). T can calculate the hash value for any input to this function. T can also carry out the XOR calculation. In fact, carrying

out the XOR calculation is an affordable capability for various RFID tags, especially for low-cost RFID tags. In our protocol, the RFID tag does not need a pseudorandom number generator PRNG. This will further reduce the cost of the RFID tags. Generally, the more power of calculating the RFID tag has, the more cost the RFID tag induces. Therefore, in our authentication protocol, the most expensive capability of the RFID tag needs is to carry out the hash function evaluation and the XOR calculation. For each RFID tag in our system, it has two memories: one is volatile memory (VM) with bit-length β, the other is non-volatile memory (NVM) with the same bit-length β. The volatile memory is used to store metadata in an RFID tag and reader authentication process. Here the metadata stored in the volatile memory of the RFID tag represents the data the tag T calculated by itself and received from reader R. The metadata stored in the VM will be automatically deleted as soon as the authentication process is fulfilled. In fact, some metadata only stays in the VM for a temporal state even before the authentication process is fulfilled. The data stored in the NVM is the synchronized secret information whose bit-length is β. For initial value of the secret element k, it is shared between the database server S and the RFID tag T. In fact, it is embedded in T when T was registered to the database server S by its business owner. After a successful authentication process, T will update k with $h_{k\oplus r}(k \oplus r)$ where r is a random element generated by R's pseudorandom number generator. For any i>2, if the synchronized secret information in NVM is k_i after the successful i-th authentication process, then after the (i+1)-th successful authentication process the synchronized secret information in NVM is $k_{i+1} = h_{k_i \oplus r_{i+1}}(k_i \oplus r_{i+1})$ where r_{i+1} is a random number generated by R's pseudorandom number generator PRNG.

In our authentication protocol, we set up a monitor M in the database server S. M is used to scrutinize whether the RFID tag T updated its secret information after interacting with an authorized RFID reader R. The detail of the structure for the database server will be introduced in subsection 3.4. The communication channel between the RFID tag T and reader R is insecure while the channel between R and the database server S is secure.

3.3 The Proposed Protocol

The mutual authentication protocol for RFID tags based on synchronized secret information with monitor is summarized in the following figure.

The details of the mutual authentication protocol are presented as:

1. RFID reader R uses the PRNG and a seed α to generate a random number r, then uses the hash function h() to get $t_1 = h(r)$. R then sends Request and t_1 to RFID tag T.
2. T first checks $t_1 \neq \overbrace{\{0,...,0\}}^{\beta}$. If yes, then T uses its secret element k and the received t_1 to get $x_1 = h_{k\oplus t_1}(k \oplus t_1)$. T then sends x_1 back to R.
3. After receiving x_1, R sends x_1 and its random number r to the database server S.

DatabaseServer (h())	Reader (PRNG, h())	Tag (h())
	r = PRNG(α) $t_1 = h(r)$	
	Request, t_1 →	check $t_1 \neq \overbrace{\{0,...,0\}}^{\beta}$ set $x_1 = h_{k \oplus t_1}(k \oplus t_1)$
	← x_1	
← x_1, r		
Identifying k through computing $g = h(k \oplus h(r))$ Checking whether $x_1 = g$, then setting $y_1 = h(g \oplus k \oplus r)$, Checking the monitor M If $M < \theta$, then		
y_1→		
	r, y_1→	
		Set $t_2 = h(r)$ Check
whether $t_1 = t_2$		
		If yes, then set $y_2 = h(k \oplus r \oplus x_1)$ If $y_1 = y_2$, then Replace k with $h_{k \oplus r}(k \oplus r)$

Fig. 1. Mutual authentication protocol for RFID tags based on synchronized secret information with monitor. In this figure, β is the bit-length of the output of the hash function h(). After the successful mutual authentication, the RFID tag will delete the old private element k and update the non-volatile memory using the new private element $h_{k \oplus r}(k \oplus r)$, where the hash value is the keyed hash evaluation with $k \oplus r$.

4. S first uses hash function h() to get $h(r)$, then identify ID of tag T through (1) finding a value k in the database to compute $g = h(k \oplus h(r))$; (2) checking whether $x_1 = g$. If there is only value k after the process, then the identification ID in the list of k is the ID of the tag T.

5. S then computes $y_1 = h(x_1 \oplus k \oplus r)$ using x_1, k and r. Before sending y_1 back to R, S checks whether the monitor M in the list of k is less than θ. If yes, then S sends y_1 back to R. Otherwise, S will inform R that the tag T may be prevented from updating its secret information k by adversaries. An order will be instantly sent to the system administrator (e.g. the business owner) to inspect the tag.
6. After receiving y_1, R forwards y_1 and its random element r to T.
7. T first sets $t_2 = h(r)$ and then checks $t_1 = t_2$. If they are not equal, then T stops here; otherwise, T sets $y_2 = h(k \oplus r \oplus x_1)$ and compares $y_1 = y_2$. If yes, then T replaces k with $h_{k \oplus r}(k \oplus r)$.
8. For the database server S, if k is found in K_{OLD}, then the monitor M will be updated as M+1 while the secret information k will keep stable for this authentication process. If k is found in K_{NEW}, then the monitor M will not be updated while the secret information k in K_{NEW} will be replaced with $h_{k \oplus r}(k \oplus r)$ and the value of k will be copied to the field K_{OLD}.

3.4 The Structure of the Database Server

The database server plays an important role in the mutual authentication protocol between the RFID tag and reader. One the one hand, the database server maintains the identification information for each registered RFID tag and stores some private elements which can be used to authenticate RFID reader to tag. On the other hand, the database server maintains a secure channel for authorized RFID readers. This enables to authenticate RFID tag to reader. For each registered RFID tag in the system, the database server needs to maintain a list. The list is composed of four components: the identification information part (ID), the old secret information part (K_{OLD}), the new secret information part (K_{NEW}), and the authentication monitor part (M). Here M is an integer that will augment if the secret element k is found in K_{OLD}; otherwise, it will keep stable. That is,

$$M = \begin{cases} M+1, & \text{if } k \text{ is found in } K_{OLD} \\ M, & \text{if } k \text{ is found in } K_{NEW} \end{cases}$$

There is also a threshold θ for M. If for the (i+1)-th authentication process, the database server identifies that $M \geq \theta$, then the server will inform the RFID reader of the the (i+1)-th authentication process that the tag (that the reader interacts with) may be compromised or prevented from updating the secret information k by malicious parties. The initial value of M is 0.

Remark 1. We suppose the first i+2 times authentication process are all successfully finished. Then we can set the M value for ID_T as 0 in the above table. Otherwise, the M value will be j (where $j<\theta$). This implies that the RFID tag T has not updated its secret element k for j times.

The internal components of the database server in the (i+3)-th authentication process for tag T

ID	K_{OLD}	K_{NEW}	M
…	…	…	…
ID_T	k	$h_{k\oplus r}(k \oplus r)$	0
…	…	…	…
…	…	…	…

Fig. 2. The structure of the database server. We use one RFID tag instance T to demonstrate the components of the field of the database server. ID_T is the unique identification number of T. ID_T keeps stable in any authentication process. Suppose in the (i+1)-th authentication process the synchronized secret element of T is k, then for the (i+2)-th authentication process the secret element in K_{OLD} is k while in K_{NEW} the secret element is $h_{k\oplus r}(k \oplus r)$ after a successful authentication process. Here r is a random element that was generated by an authorized RFID reader involving the authentication process.

4 Security Analysis and Comparison

This section will provide the security analysis and a comparison of security characteristics between our protocol and some previous protocols.

4.1 Security Analysis

Attack on user privacy: From the construction of our protocol, we can see that no identity was released by an RFID tag participating in the authentication process. Therefore, eavesdropping is not an issue as long as the hash function h() is one-way hash function so that no usable information on its pre-image is revealed. As a result, user privacy is guaranteed.

Attack on the RFID tag: If an adversary tries to impersonate an authorized RFID reader and attack on the RFID tag, then the adversary does not know the secret information k. As a result, the adversary cannot compute g and y_1. Therefore, the adversary cannot let the RFID tag accept its access. On the other hand, the communication channel between S and R is secure. This prevents the adversary from interacting with the database server S.

Mutual authentication: The authentication between R and T is mutual authentication. R is authenticated to T by checking whether $t_1 = t_2$ and $y_1 = y_2$. T is authenticated to R by checking whether $x_1 = g$. This is actually done by the database server.

Forward security and untraceable property: If an adversary obtains the secret information k by compromising the RFID tag, then the adversary cannot trace back the previous authentication the tag involved. This is because for each authentication run,

the RFID reader generates a random number r. And the secret information k is mixed and hashed with r. Therefore, forward security as well as untraceable property is preserved for successive and fulfilled authentication runs.

Anti-cloning: If an adversary tries to make cloning by replaying intercepted interactions, then the adversary cannot success cloning because r is different for each authentication run. On the other hand, the adversary cannot work out a valid x_1 without knowing the secret information k.

In order to compare the security characteristics in different RFID authentication protocols, we give the following definition:

Definition 1. DeSynchronization Detection Security: In the proposed RFID tag authentication protocol, there is a monitor M in the database field which can inform the RFID reader that the targeted RFID tag may be prevented from updating its secret information by adversaries if $M \geq \theta$. We name this characteristic as the **DeSynchronization Detection Security (DDS)**.

4.2 Security Characteristic Comparison

By the comparison studied in the following table, we can know that the protocol in our paper offers the most enhanced security in RFID mutual authentication protocols [5], [7], [11], [12], [14] and our paper.

Table 2. Security characteristic comparison

Protocol / Characteristic	[7]	Our protocol	[11]	[14]	[5]	[12]
Forward security	No	Partially satisfy	Partially satisfy	No	Partially satisfy	Partially satisfy
Mutual authentication	Yes	Yes	Yes	Yes	Yes	Yes
DDS	Not satisfy	Satisfy	Not satisfy	Not satisfy	Not satisfy	Not satisfy
Indistinguishable	Satisfy	Satisfy	Satisfy	Partially satisfy	Not satisfy	Not satisfy
Anti-cloning	Satisfy	Satisfy	Satisfy	Satisfy	Satisfy	Satisfy

5 Efficiency and Comparison

In this section, we will measure the computational efficiency of tag T. Following that, we will then compare our protocol with the existing protocols from the computational efficiency point of view.

5.1 Efficiency of Tag

The RFID tag in our protocol does not need the pseudorandom number generator PRNG. For a successful authentication process, the RFID tag needs four hash operations. However, our protocol does not need the strong assumption Hash Chain Requirement. The communication complexity and the non-volatile memory of our protocol are identical to those of [11].

5.2 Efficiency Comparison

In order to compare the efficiency of RFID tag authentication of our protocol with previous protocols, we derive a definition for hash chain requirement from a previous protocol in [11].

Definition 2. Hash Chain Requirement: Let s be the total number of RFID tags in an RFID system. Let t be the maximum number of authentication times of each RFID tag T. H() is a one-way hash function. The hash chain can be constructed: $H_1 = H(K_0)$, where K_0 is the a secret element; the second value in the hash chain is $H_2 = H(H_1)$; for any $3 \leq j \leq st$, the j-th value in the hash chain is $H_j = H(H_{j-1})$. If the secret element K_0 needs be chosen such that the hash chain length is longer than st, then we say the hash function H() together with K_0 satisfy the *Hash Chain Requirement.*

By this definition, it is easy to see that the authentication protocol in [11] employed a one-way hash function and an initial secret element which need to meet the Hash Chain Requirement. On the contrary, our protocol does not need to use a hash function which meets the strong assumption. This is because in our authentication protocol the secret element has been mixed with a different random number once whenever an authorized RFID reader interacts with the tag. Therefore, in the hash chain, the pre-image of the current hash value is not the previous hash value adjunct to the current hash value.

Table 3. Efficiency in RFID tag

Protocol / Efficiency	[14]	Our protocol	[11]	[7]
PRNG operation		0	1	1
Hash Operation	2	4	3	2
Communication complexity	3γ	4γ	4γ	4γ
Non-volatile memory	2γ	γ	γ	2γ
Hash chain requirement	_	_	+	_

Remark 2. In the above table, we use the similar measure as that of [11]. γ denotes the number of bits of a data unit. In the last row of the table, '+' represents the protocol in [11] should meet the Hash Chain Requirement; '-' represents our protocol is not restricted by the requirement. In other words, the hash function and the initial secret information in our protocol do not need to satisfy the Hash Chain Requirement.

From the above table, we can see that the protocol [11] and our protocol induces more hash operations than the protocols in [7] and [14]. The non-volatile memory cost in our protocol and [11] is identical. From the above table, both of the protocol [11] and our protocol require the non-volatile memory (50% off) less than the protocols in [7] and [14]. Notice that the non-volatile memory is a comparative expensive unit in RFID tags. Therefore, the protocols of [11] and ours have economic effective merit. On the other hand, although our protocol needs one more hash operation than that of [11], the RFID tag in our protocol does not need a PRNG and the strong assumption-Hash Chain Requirement. Therefore, based on the above comparison, our protocol is better desirable than the protocols [11].

6 Conclusions

In this paper, we proposed a mutual authentication protocol for RFID tags based on synchronized secret information with a monitor. Some characteristics are summarized as follows:

- The authentication between tag and reader is mutual authentication.
- The memory of the tag is portioned into two parts: volatile memory and non-volatile memory. Thus, it reduces the cost of tags.
- The synchronized secret information between the tag and the database is monitored so that some attacks can be detected.
- The tag does not need a pseudorandom number generator.
- The synchronized secret information is masked with a random number. Thus, the hash chain requirement is on the rampage.

References

1. Avoine, G., Oechslin, P.: A Scalable and Provably Secure Hash Based RFID Protocol. In: The 2nd IEEE International Workshop on Pervasive Computing and Communication Security, Kauai Island, Hawaii, USA, March 8th, pp. 110–114 (2005)
2. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic Approach to Privacy-friendly Tags. In: RFID Privacy Workshop, MIT, Cambridge (2003)
3. Juels, A.: Minimalist Cryptography for RFID Tags. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, Springer, Heidelberg (2005)
4. Vajda, I., Butty'an, L.: Lightweight authentication protocols for low-cost RFID tags. In: 2nd Workshop on Security in Ubiquitous Computing (2003)
5. Henrici, D., Műller, P.: Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers. In: Workshop on Pervasive Computing and Communications Security (2004)

6. Avoine, G., Oechslin, P.: RFID Traceability: A Multilayer Problem. Financial Cryptography (2005)
7. Molnar, D., Wagner, D.: Privacy and Security in Library RFID Issues, Practices, and Architectures. In: ACM Conference on Computer and Communication Security (2004)
8. Feldhofer, M., Dominikus, S., Wolkerstorfer, J.: Strong Authentication for RFID Systems Using the AES Algorithm. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, Springer, Heidelberg (2004)
9. Sean, W., Thomas, L.: Automatic identification and data collection technologies in the transportation industry: Barcode and RFID. Technical Report (2001)
10. Sarma, S.E., Weis, S.A., Engels, D.W.: RFID systems, security and privacy implications. In: Technicalreport MIT-AUTOID-WH-014, AutoID center, MIT, Cambridge (2002)
11. Lee, S., Lee, H., Asano, T., Kim, K.: Enhanced RFID Mutual Authentication Scheme based on Synchronized Secret Information. AUTO-ID Labs White Paper, WP-HARDWARE-032 (2006)
12. Dimitriou, T.: A Lightweight RFID Protocol to Protect against Traceability and Cloning Attacks. In: International Conference on Security and Privacy for Emerging Areas in Communication Networks- SecComm (September 2005)
13. Jules, A., Rivest, R., Szydlo, M.: The Blocker Tag: Selective Blocking of RFID tags for Consumer Privacy. In: ACM Conference on Computer and Communication Security – ACM CCS, pp. 103–111 (October 2003)
14. Lee, S.M., Hwang, Y.J., Lee, D.H., Lim, J.I.: Efficient Authentication for Low-Cost RFID Systems. In: International Conference on Computational Science and Its Applications – ICCSA, pp. 619–627 (May 2005)
15. Choi, E.Y., Lee, S.M., Lee, D.H.: Efficient RFID Authentication Protocol for Ubiquitous Computing Environment. In: Proceedings of International Workshop on Security in Ubiquitous Computing Systems (December 2005)
16. Tuyls, P., Batina, L.: RFID-tags for Anti-Counterfeiting. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 115–131. Springer, Heidelberg (2006)

Non-linear Least Squares Features Transformation for Improving the Performance of Probabilistic Neural Networks in Classifying Human Brain Tumors on MRI

Pantelis Georgiadis[1], Dionisis Cavouras[2], Ioannis Kalatzis[2], Antonis Daskalakis[1], George Kagadis[1], Koralia Sifaki[3], Menelaos Malamas[3], George Nikiforidis[1], and Ekaterini Solomou[4]

[1] Medical Image Processing and Analysis Group (M.I.P.A.), Laboratory of Medical Physics, School of Medicine, University of Patras, Rio, GR-26500 Greece, pgeorgiadis@med.upatras.gr
[2] Medical Signal and Image Processing Lab, Department of Medical Instruments Technology, Technological Educational Institute of Athens, Ag. Spyridonos Street, Aigaleo, 122 10, Athens, Greece
[3] 251 General Hellenic Airforce Hospital, MRI Unit, Katehaki, Athens, GR-11525, Greece
[4] Department of Radiology, School of Medicine, University of Patras, Rio, GR-26503 Greece

Abstract. The aim of the present study was to design, implement, and evaluate a software system for discriminating between metastases, meningiomas, and gliomas on MRI. The proposed classifier is a modified probabilistic neural network (PNN), incorporating a second degree least squares features transformation (LSFT) into the PNN classifier. Thirty-six textural features were extracted from each one of 75 T1-weighted post-contrast MR images (24 metastases, 21 meningiomas, and 30 gliomas). Classification performance was evaluated employing the leave-one-out method and for all possible textural feature combinations. LSFT enhanced the performance of the PNN, achieving 93.33% in discriminating between the three major types of human brain tumors, against 89.33% scored by the PNN alone. Best feature combination for achieving highest discrimination power included the mean value and entropy, which reflect specific properties of texture, i.e. signal strength and inhomogeneity. LSFT improved PNN performance, increased class separability, and resulted in dimensionality reduction.

1 Introduction

One of the most promising techniques for generating useful information for brain tissue characterization is Magnetic Resonance Imaging (MRI). MRI, in comparison with other diagnostic imaging modalities, such as computerized tomography, provides superior contrast and resolution for different brain tissues [1]. Additionally, MR images encapsulate valuable information regarding numerous tissue parameters (proton density, spin-lattice (T1) and spin–spin (T2) relaxation times, flow velocity, and chemical shift), which lead to more accurate brain tissue characterization. These unique advantages have characterized MRI as the method of choice in brain tumor studies [2]. In order to extract the diagnostic information of different parameters reflected in MRI, image analysis techniques have been employed.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, pp. 239 – 247, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Several studies have proposed various approaches to characterize human brain tumors, using either MR textural [3-5], spectroscopic [6-10], or a combination of both textural and spectroscopic parameters [11-12]. These studies have formulated the discrimination between several tumor types as a two-class problem, employing either statistical analysis (linear discriminant analysis, analysis of variance, hierarchical ascending classification) or state of art classifiers, such as support vector machines.

The aim of the present study was to improve tumor classification accuracy, as compared to previous studies, in distinguishing between three major categories of human brain tumors: metastases, meningiomas, and gliomas. We considered a three-class problem formulation, we employed textural features, and we conditioned these features by means of least squares features transformation (LSFT) to boost the performance of the Probabilistic Neural Network (PNN) classifier.

2 Materials and Methods

2.1 Data Acquisition

A total number of 75 T1-weighted post-contrast (Gadolinium) MR images were obtained from the Hellenic Airforce Hospital with verified intracranial tumors, using a SIEMENS-*Sonata* 1.5 Tesla MR Unit. The image dataset comprised 24 metastases, 21 meningiomas, and 30 gliomas. From each case, only T1-weighted post-contrast (Gadolinium) images, with Spin Echo (SE) sequence, Echo Time (TE = 15ms), and Repetition Time (TR = 500ms), were used for further analysis. Transverse images were selected through the tumor's center by an expert radiologist (M.M.).

2.2 Feature Extraction

Utilizing these images, the radiologist specified Regions Of Interest (ROIs) that included tumor regions, using custom developed software (Fig. 1). From each ROI, a series of 36 textural features were generated; 4 features where extracted from the ROI's histogram, expressing information concerning the frequency of appearance of

Table 1. Textural features extracted

Methods	Features
Histogram (1st order statistics)	Mean Value, Standard Deviation, Skewness, Kurtosis
Mean and range of 0°, 45°, 90° and 135° co-occurrence matrices (2nd order statistics)	Angular Second Moment, Contrast, Correlation, Sum Of Squares, Inverse Difference Moment, Sum Average, Sum Variance, Sum Entropy, Entropy, Difference Variance, Difference Entropy
Mean and range of 0°, 45°, 90° and 135° run-length matrices (2nd order statistics)	Short Run Emphasis, Long Run Emphasis, Gray Level Non Uniformity, Run Length Non Uniformity, Run Percentage

each gray level; 22 features were extracted from the gray level co-occurrence matrices [13], reflecting the frequency of appearance of each gray level at adjacent pixels; 10 features were extracted from the run-length matrices [14], describing the frequency of appearance of a set of consecutive pixels having the same gray value (Table 1).

All features were normalized to zero mean and unit standard deviation [15], according to relation (1)

$$x_i' = \frac{x_i - m}{std} \ , \tag{1}$$

where x_i and x_i' are the i-th feature values before and after the normalization respectively, and m and std are the mean value and standard deviation of feature x over all patterns (from all three tumor categories) respectively. Extracted feature vectors were fed into the classifier.

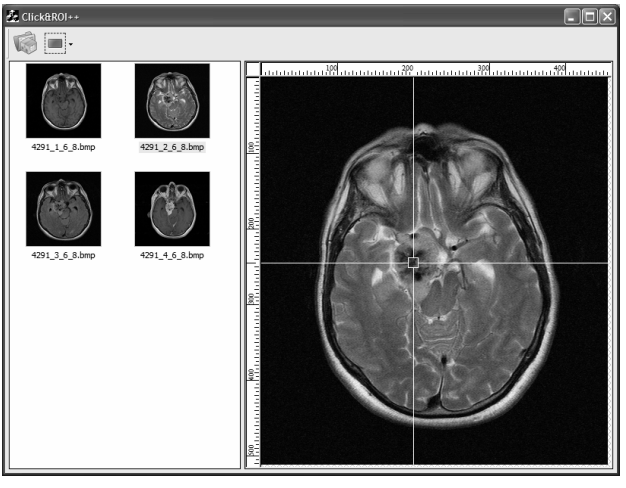


Fig. 1. Custom developed software used to extract ROIs and textural features from the MR images

2.3 Least Squares Feature Transformation – Probabilistic Neural Network Classifier (LSFT-PNN)

Due to the small size of our dataset the probabilistic neural network (PNN) classifier was chosen. The PNN is a non-parametric feed-forward neural network classifier, with low computational demands and fast in training. The discriminant function of PNN [16] is given by:

$$g_i(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}\sigma^d} \frac{1}{N_i} \sum_{j=1}^{N_i} \exp\left[-\frac{(\mathbf{x} - \mathbf{x}_{ij})^{\mathrm{T}} (\mathbf{x} - \mathbf{x}_{ij})}{2\sigma^2} \right], \tag{2}$$

where $\mathbf{x}$ is the pattern to be classified, $\mathbf{x}_{ij}$ are the training patterns, σ is the spread of the Gaussian activation function, N_i is the number of training patterns in class i, and d is the dimensionality of pattern vectors.

Training patterns $\mathbf{x}_{ij}$, prior to entering the PNN classifier, were transformed by means of a non-lineal least squares feature transformation (LSFT) technique, to render classes more separable by clustering the patterns of each class around arbitrary pre-selected points. The proposed LSFT method is an extension of the linear least squares mapping technique, introduced by [17]. Initially, pattern vectors were extended with second degree elements. Accordingly, if $\mathbf{x} = [x_1\ x_2\ \ldots\ x_d]$ is a pattern vector, where d is the input space dimensionality, then vector $\mathbf{x}$ was extended with the second degree elements x_i^2 and x_ix_j, where $i, j = 1, 2, \ldots, d$ and $i \neq j$, resulting in the following extended pattern vector:

$$\hat{\mathbf{x}} = [x_1\ x_2 \ldots\ x_d\ \ x_1^2\ \ x_2^2 \ldots x_d^2\ \ x_1x_2\ \ x_1x_3 \ldots x_1x_d\ \ x_2x_3\ x_2x_4 \ldots x_2x_d\ \ \ldots\ \ x_{d-1}x_d] \tag{3}$$

The dimensionality of the extended pattern vector ($\hat{\mathbf{x}}$) is equal to [15]:

$$\hat{d} = \frac{(d+2)!}{d!2!} - 1 \cdot \tag{4}$$

For the formulation of the LSFT 3-class problem, let space **S,** with dimensionality equal to the number of classes (K=3), and let $\mathbf{P}_i = [p_{i1}\ p_{i2}\ p_{i3}]$, i=1,2,3, be arbitrary defined points in space **S**, corresponding to each class i. A transformation $\mathbf{T}$ is sought such that the total mean square error between the transformed extended vectors ($\mathbf{T}\hat{\mathbf{x}}_{ij}$) and $\mathbf{P}_i$ is minimized as follows:

$$\nabla_{\mathbf{T}}\left[\sum_{i=1}^{K}\left(\frac{1}{N_i}\sum_{j=1}^{N_i}\left(\mathbf{T}\hat{\mathbf{x}}_{ij}-\mathbf{P}_i\right)'\left(\mathbf{T}\hat{\mathbf{x}}_{ij}-\mathbf{P}_i\right)\right)\right]=0\,, \tag{5}$$

where K is the number of classes, N_i is the number of patterns of class i, and $\hat{\mathbf{x}}_{ij}$ are the 2nd degree extended training patterns of class i.

Assuming equal *a-priori* probabilities for each class i, relation (5) results to:

$$\mathbf{T}=\left[\sum_{i=1}^{K}\left(\frac{1}{N_i}\sum_{j=1}^{N_i}\mathbf{P}_i\hat{\mathbf{x}}_{ij}'\right)\right]\left[\sum_{i=1}^{K}\left(\frac{1}{N_i}\sum_{j=1}^{N_i}\hat{\mathbf{x}}_{ij}\hat{\mathbf{x}}_{ij}'\right)\right]^{-1}. \tag{6}$$

Transformation matrix $\mathbf{T}$ is a $K\times\hat{d}$ matrix, so the decision space dimensionality is equal to the number of brain tumor classes.

Following the LSFT procedure, patterns $\hat{\mathbf{x}}_{ij}$ were fed into the PNN classifier, resulting in the final discriminant function of the LSFT-PNN classifier:

$$g_i(\mathbf{x}) = \frac{1}{(2\pi)^{\hat{d}/2}\sigma^{\hat{d}}}\frac{1}{N_i}\sum_{j=1}^{N_i}\exp\left[-\frac{(\mathbf{T}\hat{\mathbf{x}}-\mathbf{T}\hat{\mathbf{x}}_{ij})'(\mathbf{T}\hat{\mathbf{x}}-\mathbf{T}\hat{\mathbf{x}}_{ij})}{2\sigma^2}\right]. \tag{7}$$

2.4 Feature Selection and System Evaluation

Best features combination was determined employing the robust but time consuming exhaustive search method [15], which involves designing the classifier by means of every possible feature-combination, each time evaluating the classifier's performance and finally selecting that feature combination that demonstrated the highest classification accuracy with the smallest number of features.

Classifier performance was evaluated employing the leave-one-out (LOO) method [15]. Accordingly, the LSFT-PNN classifier was designed by all but one pattern-vector, which was considered as unknown and it was classified. The process was repeated, each time leaving-out a different pattern-vector, until all pattern-vectors were thus classified to one of three classes. In this way, the classifier was evaluated by pattern-vectors not involved in its design. This type of classifier training required several hours of processing time.

All algorithms were implemented using custom software developed using Microsoft's Visual C++ 6. The feature generation and classification tasks were performed on a typical desktop computer (Pentium 4 @ 3GHz with 1GB of RAM) running Windows 2000 Professional operating system.

3 Results and Discussion

The LSFT-PNN classification scheme was optimized with respect to parameter settings and available feature data. The spread of Gaussian function in equation (7) was experimentally set equal to σ=0.3. Best feature vector, used for the optimal design of the LSFT-PNN classifier, comprised the mean value, the mean entropy, and the mean and range values of the difference entropy (see Fig. 2). These features reflect the signal strength and degree of inhomogeneity of the ROIs' gray-tones. The overall classification accuracy for directly discriminating the three types of human brain tumors was 93.33% (Table 2). Figure 2 shows the scatter diagram of the transformed optimum LSFT features combination, with black-filled labels indicating misclassified patterns.

Table 2. LSFT-PNN truth table for three-class problem

	Metastases	**Meningiomas**	**Gliomas**	**Accuracy (%)**
Metastases	21	1	2	87.50
Meningiomas	1	20	0	95.24
Gliomas	1	0	29	96.67
				93.33

The performance of the LSFT-PNN algorithm was tested against that of the PNN classifier (σ=0.3), used in the LSFT-PNN technique, as well as with the Quadratic Bayesian (QB) [15] and the kNN (k=3) [15] classifiers. The classifiers were trained in a similar manner to the LSFT-PNN classifier, employing in its design the LOO procedure and the exhaustive search method. The best feature combination of the

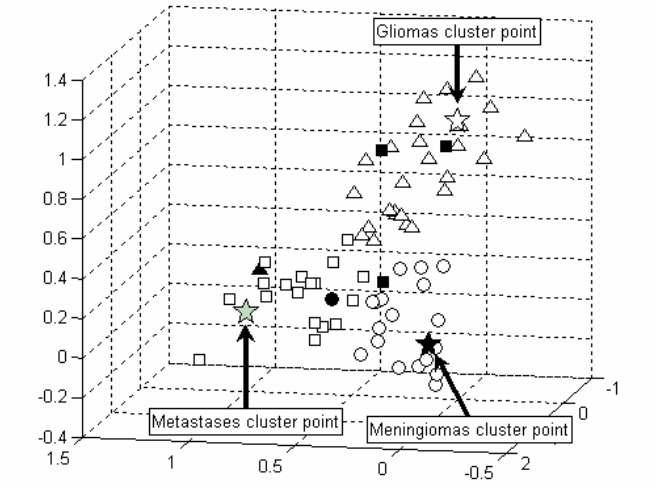


Fig. 2. Scatter diagram in the decision space of the optimum LSFT feature combination (metastasis ( ), meningiomas (○) and gliomas (Δ) classes; misclassified patterns of LSFT-PNN classification: ■, ● and ▲ respectively)

Table 3. PNN truth table for 3-class problem

	Metastases	**Meningiomas**	**Gliomas**	**Accuracy (%)**
Metastases	20	2	2	83.33
Meningiomas	2	19	0	90.48
Gliomas	2	0	28	93.33
				89.33

PNN was found to be the mean and range values of the correlation, the sum average, and the difference entropy. This features express measures of the gray-tone linear dependencies (correlation) [13], of the disperse of intensity values (sum average), and give an indication of the degree of inhomogeneity in the gray-tones of the ROIs (entropy). The performance of the PNN classification scheme is summarized in (Table 3). The overall accuracies of the QB and the kNN classifiers were 83.58% and 86.56% respectively. Best feature vector for QB was mean value, entropy, contrast, and sum of average and for kNN was angular second moment, sum of average, different entropy and short run emphasis.

The effectiveness of the LSFT transformation on the PNN's classification accuracy was also tested by formulating three, 2-class discrimination tests, as shown in Table 4. As it can be observed, the LSFT transformation improved the performance of the PNN in all three problems. Judging from the results achieved in all tests conducted (Tables 2-4), it can be claimed that the LSFT-PNN outperforms the PNN. This may be attributed to the increased class separability that the LSFT procedure provides

Table 4. LSFT-PNN versus PNN overall accuracies for 2-class problems

	Metastases vs Meningiomas (%)	Gliomas vs Meningiomas (%)	Gliomas vs Metastasis (%)
LSFT-PNN	97.50	100.00	93.75
PNN	90.00	97.83	91.67

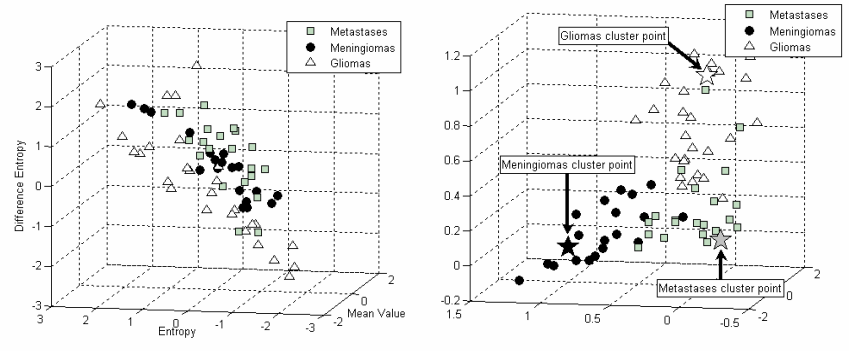


Fig. 3. Pattern vectors scatter diagram of best 3-feature combination, a/ before LSFT application and b/ after LSFT

(Fig. 3). Another advantage of the LSFT is the dimensionality reduction to the number of classes, which leads to more robust classification, independently of the number of features in the pre-transformed data.

The computational requirements of the LSFT-PNN classifier are comparable to those of the PNN, as the time required to perform the LSFT is gained in the classification step, due to the reduced feature dimensionality.

Only few studies were found in literature that employed pattern recognition techniques and texture analysis to classify human brain tumors. In a recent study [12], an SVM-based classification system achieved 95% overall accuracy in discriminating between gliomas and meningiomas. These results are comparable with the findings of the present study regarding the two-class problem formulation of these two types of brain tumors (Table 4). However, in another study [5], employing hierarchical ascending classification with correspondence factorial analysis, discrimination accuracy between different tumor types ranged between 49% (tumors Vs oedemas) and 63% (benign Vs malignant tumors), employing textural features related to gray-level homogeneity (gray-level distribution) and the existence of structures within the lesion (run length distribution). In an older study [3], discriminant analysis and the k-nearest neighbor classifier were used to differentiate between human brain tumor and oedematous tissue, achieving maximum overall accuracy equal to 95% and using textural features associated with dependencies between neighboring gray levels (correlation and run length mean gray level). Similar features were also found to give optimum discrimination accuracies in the present work.

4 Conclusion

The aim of this study was to design, implement, and evaluate a software pattern recognition system in discriminating between three categories of human brain tumors using the LSFT-PNN classification algorithm. Least squares features transformation was proved to improve the performance of the PNN classifier, mainly due to the increased class separability and to dimensionality reduction that the LSFT procedure provides.

Acknowledgments. Funding by the University of Patras Research Committee under the basic research program K. Karatheodoris, Project title "Computer Assisted Diagnosis of Brain Tumors based on Statistical Methods and Pattern Recognition Techniques" is gratefully acknowledged.

References

1. Shen, S., Sandham, W., Granat, M., Sterr, A.: MRI fuzzy segmentation of brain tissue using neighborhood attraction with neural-network optimization. IEEE Trans Inf Technol Biomed. 9, 459–467 (2005)
2. Soltanian-Zadeh, H., Peck, D.J., Windham, J.P., Mikkelsen, T.: Brain tumor segmentation and characterization by pattern analysis of multispectral NMR images. NMR Biomed. 11, 201–208 (1998)
3. Lerski, R.A., Straughan, K., Schad, L.R., Boyce, D., Bluml, S., Zuna, I.: MR image texture analysis–an approach to tissue characterization. Magn Reson Imaging 11, 873–887 (1993)
4. Schad, L.R., Bluml, S., Zuna, I.: MR tissue characterization of intracranial tumors by means of texture analysis. Magn Reson Imaging 11, 889–896 (1993)
5. Herlidou-Meme, S., Constans, J.M., Carsin, B., Olivie, D., Eliat, P.A., Nadal-Desbarats, L., Gondry, C., Le Rumeur, E., Idy-Peretti, I., de Certaines, J.D.: MRI texture analysis on texture test objects, normal brain and intracranial tumors. Magn Reson Imaging 21, 989–993 (2003)
6. Cho, Y.-D., Choi, G.-H., Lee, S.-P., Kim, J.-K.: ^{1}H-MRS metabolic patterns for distinguishing between meningiomas and other brain tumors. Magnetic Resonance Imaging 21, 663–672 (2003)
7. Tate, A.R., Majos, C., Moreno, A., Howe, F.A., Griffiths, J.R., Arus, C.: Automated classification of short echo time in in vivo ^{1}H brain tumor spectra: A multicenter study. Magn Reson Med. 49, 29–36 (2003)
8. Devos, A., Lukas, L., Suykens, J.A., Vanhamme, L., Tate, A.R., Howe, F.A., Majos, C., Moreno-Torres, A., van der Graaf, M., Arus, C., Van Huffel, S.: Classification of brain tumours using short echo time ^{1}H MR spectra. J Magn Reson. 170, 164–175 (2004)
9. Lukas, L., Devos, A., Suykens, J.A., Vanhamme, L., Howe, F.A., Majos, C., Moreno-Torres, A., Van der Graaf, M., Tate, A.R., Arus, C., Van Huffel, S.: Brain tumor classification based on long echo proton MRS signals. Artif Intell Med. 31, 73–89 (2004)
10. Menze, B.H., Lichy, M.P., Bachert, P., Kelm, B.M., Schlemmer, H.P., Hamprecht, F.A.: Optimal classification of long echo time in vivo magnetic resonance spectra in the detection of recurrent brain tumors. NMR Biomed. 19, 599–609 (2006)

11. Simonetti, A.W., Melssen, W.J.: Combination of feature-reduced MR spectroscopic and MR imaging data for improved brain tumor classification. NMR Biomed. 18, 34–43 (2005)
12. Devos, A., Simonetti, A.W., van der Graaf, M., Lukas, L., Suykens, J.A., Vanhamme, L., Buydens, L.M., Heerschap, A., Van Huffel, S.: The use of multivariate MR imaging intensities versus metabolic data from MR spectroscopic imaging for brain tumour classification. J Magn Reson. 173, 218–228 (2005)
13. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Trans Syst Man Cybern. SMC 3, 610–621 (1973)
14. Galloway, M.M.: Texture analysis using grey level run lengths. Comp. Graph. and. Image Proc. 4, 172–179 (1975)
15. Theodoridis, S., Koutroumbas, K.: Pattern recognition. Academic Press, New York (1999)
16. Specht, D.F.: Probabilistic neural networks. Neural Networks 3, 109–118 (1990)
17. Ahmed, N., Rao, R.: Orthogonal transforms for digital signal processing. Springer, Heidelberg (1975)

Adaptive Scheduling for Real-Time Network Traffic Using Agent-Based Simulation

Moutaz Saleh[1] and Zulaiha Ali Othman[1]

[1] Faculty of Science & IT, University Kebangsaan Malaysia,
43600 UKM Bangi, Selangor
{msalleh, zao}@ftsm.ukm.my

Abstract. Since several years, communication networks have known a surprising growth. The increased number of users, the consequent increase of traffic, and the request for new services involve the development of new technologies and the deployment of high throughput networks. Networking technology has correspondingly grown to meet the diverse needs of applications and network administration. In response to the complexity of communication networks, simulation was and remains the only way to evaluate network performance. Unfortunately, traditional simulation methods are not adapted to all networks such as networks with quality of service and networks with dynamic aspect. To overcome this limitation, a new method to simulate dynamic networks based on multi-agents simulation and behavioral approach had been proposed. In this paper, we present an adaptive approach to schedule real-time network traffic using the agent based simulation concept. The paper introduces an adaptive real-time agent scheduler (ARTAS) architecture and agent model as basis for scheduling real-time packet networks.

Keywords: Agents, Simulation, Scheduling, Real-time network.

1 Introduction

There has been considerable recent interest in agent-based systems [1],[2],[3],[4],[5]; systems based on autonomous software and/or hardware components (agents) which cooperate within an environment to perform some task. An *agent* can be viewed as a self-contained, concurrently executing thread of control that encapsulates some state and communicates with its environment and possibly other agents via some sort of message passing [6]. Agent-based systems offer advantages when independently developed components must interoperate in a heterogeneous environment, e.g., the INTERNET, and agent-based systems are increasingly being applied in a wide range of areas including telecommunications, business process modeling, computer games, control of mobile robots and military simulations [7]. Agents are emerging as one of the key technologies which will facilitate the transition to and the implementation of the Information Society and, as such, they can have enormous long-term benefits to society.

While agents offer great promise, adoption of this new technology has been hampered by the limitations of current development tools and methodologies. Multi-agent systems are often extremely complex and it can be difficult to formally verify

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, pp. 248 – 261, 2007.
© Springer-Verlag Berlin Heidelberg 2007

their properties [8]. As a result, design and implementation remains largely experimental, and experimental approaches are likely to remain important for the foreseeable future. In this context, simulation has a key role to play in the development of agent-based systems, allowing the agent designer to learn more about the behavior of a system or to investigate the implications of alternative architectures, and the agent researcher to probe the relationships between agent architectures, environments, and behavior [9]. The use of simulation allows a degree of control over experimental conditions and facilitates the replication of results in a way that is difficult or impossible with a prototype or fielded system, and it allows the agent designer or researcher to focus on a particular aspect of the system, deferring problems that are not central to the research or that are beyond the capabilities of current AI technology. Hence, simulation has traditionally played an important role in agent research and a wide range of test beds have been developed to support the design and analysis of agent architectures and systems.

In fact, multi-agent simulation derives from three strong ideas: to use models centered on entities and their interactions rather than on relations between measured values, to consider that the general dynamicity of the system results from interactions between these entities, and to make the assumption that agents are able to describe the behavior of the entities.

2 Real-Time Systems

A real-time system is typically composed of several or sequential tasks with timing constraints. In most real time systems, tasks are invocated repeatedly: each invocation of a task is referred as a job; and the corresponding time of invocation is referred as the job's release time or job's deadline [10]. Thus, the relative deadline parameter is used to specify the timing constraints of the jobs.

A real-time system has two notions of correctness: logical and temporal [10]. In particular, in addition to producing correct outputs (logical correctness), such a system needs to ensure that these outputs are produced at the correct time (temporal correctness). However, selecting appropriate methods for scheduling activities is one of the important considerations in the design of a real-time system [11]; such methods are essential to ensure that all activities are able to meet their timing constraints. These timing constraints are usually specified using a deadline, which corresponds to the time by which a specific operation must complete.

Real-time systems can be broadly classified as hard or soft depending on the criticality of deadlines [12]. In hard real-time systems, all deadlines must be met; equivalently, a deadline miss results in an incorrect system. On the other hand, in a soft real-time system, timing constraints are less stringent; occasional deadline misses do not affect the correctness of the system.

3 Related Work

To manage and promote disciplined development of agent-based systems, a comprehensive and detailed architecture is required. For such purpose, the past

decades witnessed several proposed architectures with particular attention on the following architectures:

A. AI-Based Approaches: A variety of agent architectures has been proposed and employed with the emphasis on decoupling agents' actions [13, 14, 15, and 16]. Such designs can be characterized as deliberative, reactive, and hybrid combinations of the two. The deliberative architecture promotes local and collective control, decision making, and execution based on declarative knowledge. The class of reactive architectures is based on building a hierarchy of tasks ranging from primitive to complex, with the former having precedence over the latter. The remaining class of architectures is known as hybrid with the obvious interpretation. For example, the architecture proposed in [13] is composed of three layers: reactive, planning, and modeling. The reactive layer is used to respond to simple needs of agents in the spirit of the reactive architectures. The other two layers are supposed to support other needs of an agent representing cognitive state of agents, reasoning about actions, and communicating their decisions to the reactive layer.

B. Distributed Software Object-Based Approaches: The basis for such architecture is an object model accounting for abstraction, encapsulation, modularity, and hierarchy [17]. The object model provided the foundation to support the fundamental extensibility, scalability, management, and persistence traits for software architecture. The unified modeling language (UML) provides a set of modeling concepts and constructs. With UML diagrams, one can capture various static and dynamic aspects of a system.

C. Layered Architecture: This architecture consists of seven layers, network (the lowest layer), middleware, simulation, modeling, search, decision, and collaboration, where the lower layers provide services for the upper layers. Higher layers can use and possibly subsume the roles of the lower layers. Each layer is considered in isolation and in relation to others [18].

4 Multi-agent Simulation Process

In a discrete event simulation the time and nature of future events is computed in a predetermined fashion from the list of past events which have occurred. Thus the designer of a simulation will typically pre-specify all possible transitions, and will not leave the definition of state transitions to entities which are not fully pre-determined. Thus it would be very useful to introduce agents in a simulation whose behavior is determined by specific circumstances and through adaptive behavior. The alternative we propose is that, in addition to the usual attributes of a discrete event simulation (such as event lists and possibly random number generator driven events), a simulation should contain a certain number of agents. These agents store information during a simulation, and use it to modify their behavior during that same simulation or during distinct simulation runs.

From the point of view of agent-based simulation, we break up our simulation into three stages as shown in Fig. 1:

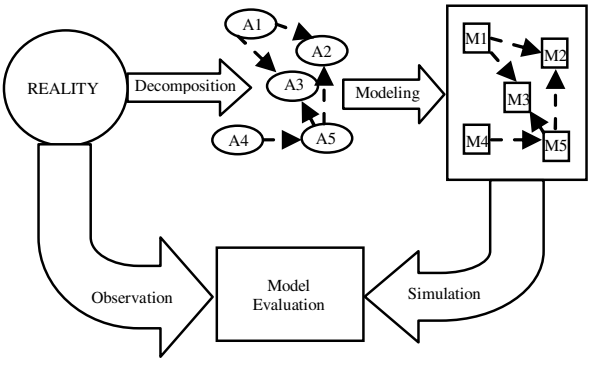


Fig. 1. Agent-Based Simulation Stages

- First stage: includes the decomposition of the real phenomenon into a set of autonomous elements that interacts between each other, and whose interactions reproduce the real phenomenon.
- Second stage: includes modeling each element by an agent, and definition of its knowledge, its functional capacities, its behaviors and its interaction modes with other agents.
- Third stage: includes the description of possible actions between agents, by defining the environment in which these agents evolve and the rules which control them.

By taking again the three stages described in Fig. 1, we place ourselves in an agent context. Thus, we follow the steps of a multi-agents simulation process:

- System Decomposition: a queuing model is an infrastructure that gathers six types of elements: server, scheduler, queue, source, packet, and clock. In addition to the model itself, we have to create a new element (Main) that will be the principle actor of the system behavior.
- Modeling each element by an agent: with regard to the elements of the environment, we consider two types of elements: static (passive) agents and dynamic (active) agents. To bring great dynamicity to the system and preserve computer resources we decide to model server, scheduler, queue, and main as active agents while source, packet, and clock as passive agents.
- Description of possible actions between the agents: there are many possible actions between the agents, i.e. messages exchange between the agents to take certain action, change their policy, or update their information.

5 ARTAS Architecture

Our Adaptive Real-Time Agent Scheduling (ARTAS) architecture is a three-layered architecture as depicted in Fig.2. At the lowest layer, we assume having a real-time

operating system (RTOS). Above that are the real-time middleware services which consist of the real-time scheduling agent and the real-time queue agent. At this layer, all tasks are scheduled based on the implemented algorithm, i.e. in our case the Hierarchical Diff-EDF scheduler. Finally, the real-time agent services layer extends the scheduling services resulting in task completion. Moreover, the top layer provides a service for real-time agent communications and interactions.

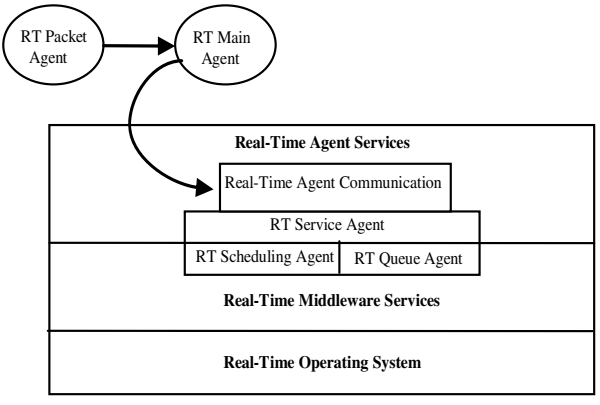


Fig. 2. ARTAS Architecture

RT Packet Agent: communication among agents in this architecture is performed through having a packet request for service originated by the packet agent and coordinated by the Main agent. Each packet request has a formal description of <*A*, *I*, *D* >, where *A* represents the source to generate this packet request, *I* is the flow priority to which this packet belongs, and *D* represents the deadline by which the request packet must be completed. If the servicing agent cannot complete this request before its deadline expires then it will be discarded.

RT Main Agent: as we mentioned earlier, the Main agent is the principle actor of the system behavior. Hence, all packet requests should be sent through this agent. It is the agent to control the entire environment through communications with the other agents in the system. The Main agent enforces other agents' policies, disciplines, and actions.

RT Service Agent: is an active agent which is responsible for packet request completion. It is the Main agent which tells this agent when to change their service discipline. The service agent keeps track of the missed packet and reported them to the Main agent.

RT Scheduler Agent: our real-time agent scheduling algorithm performs schedulability based on the EDF algorithm. The packet request is scheduled depending on their original flow, i.e. RT or Non-RT flow. The scheduler agent treats the incoming packets to achieve the best QoS flow's requirements.

RT Queue Agent: this is an active agent which treats the incoming packets and place them in the appropriate defined queue based on their flow characteristics requested by the Main agent. The queues have limited size and provided with threshold. The queue agent monitors the queue threshold and interacts with the Main agent to inform the queues' status. It also, has the authority to control the queues' filling rate in real-time and the number of discarded packets according to their class.

6 Agent Model

Our RT Agent model allows for the expression and enforcement of timing constraints on real-time agent interactions. The model is based on the assumption that agents may be able to perform their tasks in variety ways. It is made up of real-time agents (RTAgent) and a set of communications among the real-time agents. Fig. 3 displays the active elements of the model.

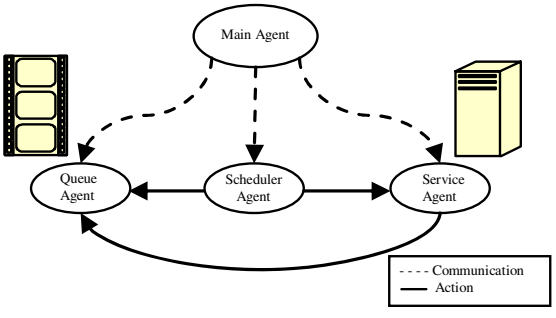


Fig. 3. Agent Model

7 System Agents Implementation

The system is modeled as agents and their interactions rather than relations based on calculated measures; i.e. the system behaves on dynamic basis rather than static one. The agents in the system are these entities capable of describing it own behavior with other entities within the system and with the environment i.e., constraints specifying dynamicity. The phenomenon (real system) is modeled by composing the agents shown in the agent model. For each agent, we identify the definition of its knowledge, its functional capabilities, its behavior and its interaction modes with other agents.

7.1 Main Agent

The main agent is the principle actor of the system behavior. It monitors the other agents and provides them with the directions for use. It decides changes of behavior, treats the messages coming from the other agents. The main agent takes decisions, has learning and high communication level capabilities and can provide other agents with

information at any time during the simulation. The main agent has several functional capabilities such as setting the system environment, managing packets' request and generating reports. Fig. 4 shows the packet request process handled by the main agent. The process starts when the Main agent accepts the request. At that time the Main agent will interact with the scheduler agent to inform a new request arrival. As a part of the operation, the scheduler has to communicate with server agent to check his status. If the server agent replies with idle then the request will immediately served and reported to the Main agent. Otherwise, if replies with busy, the scheduler agent has to communicate with the queue agent requesting to enqueue the packet. The queue agent will execute the request, based on the packet QoS specifications, and the action will be acknowledged to the scheduler agent.

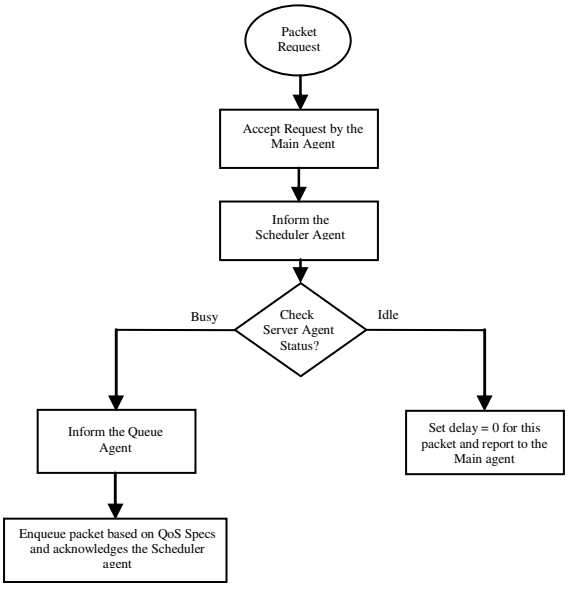


Fig. 4. Packet Request Process

7.2 Scheduler Agent

The scheduler agent is the key behind enforcing expressed timing and QoS constrains. It plays the main role in determining the disciplines of how the packets are going to be served. For our scheduler agent, two disciplines, in hierarchical arrangement, have been defined: the Real-Time (RT) such as EFD used to schedule multimedia traffic and the None-Real-Time (NRT) such as FCFS used to schedule the best effort traffic. The main tasks this agent has to perform as a part of our model implementation include packet classification, packet deadline adjustment, and monitoring system events such as arrival and departure. For example, Fig. 5 shows the packet classification process being handled by the scheduler agent.

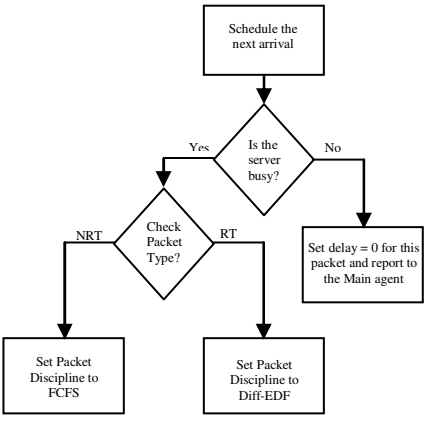


Fig. 5. Packet Classification Process

7.3 Queue Agent

This agent is the system queues' manager. It is responsible for treating the incoming packets, when the server agent is busy, by placing them into the appropriate defined system queue. To complete such a task, the queue agent performs an enqueuing operation; where the EDF queue is used for the real-time packets while the FCFS queue is used for the Non real-time packets. On the other hand, the queue agent performs a dequeuing operation whenever the server agent requests an enqueued packet to get ready for service.

Enqueuing Operation: to implement such operation, we use the Linked List data structure. For FCFS queue, we use ordinary linked list as packet arrangement is based only on the inter-arrival time of each packet. Consequently, packets are enqueued with smallest inter-arrival time at the head of the queue. On the other hand, the EDF queue is implemented using Sorted Linked List, where sorting is based on the min value of packet deadline and inter-arrival time. The reason to choose this type of Linked List is to reduce scheduler complexity, so that rather than the scheduler will spend the time in picking up the shortest lead time packet to get serve, the queue is ready to be served starting from the queue head.

Dequeuing Operation: this operation allows the queue agent to keep track of all packets leaving the queues to be served. The implementation of this operation is simply carried out through always picking up the packet on the head of the requested queue, identified by the server agent, to be served. Such implementation shows higher system utilization.

For the purpose of managing hierarchical scheduling, our system features a feedback control mechanism that detects overload conditions and modifies packet priority assignment accordingly. This feature is handled by the queue agent through keep observing any overload conditions at the FCFS queue. The overload condition is detected whenever the FCFS queue size exceeds its defined threshold value. If the overload condition is detected, the queue agent immediately interacts with the Main

agent to inform the condition. At this point, the Main agent changes the server agent policy, to serve the FCFS queue, until system stability. The key behind enforcing the threshold is to prevent packet starving which consequently improves the system performance.

7.4 Clock Agent

This is the system actor of recording the events occurrence. It uses the Next-Event Time Advanced approach, shown in Fig. 6, to advance the simulation clock to the time of occurrence of the system events, where at this point the state of the system is updated to account for the fact that an event has occurred. This process of advancing the simulation clock from one event time to another is continued until eventually some pre-specified stopping condition is satisfied.

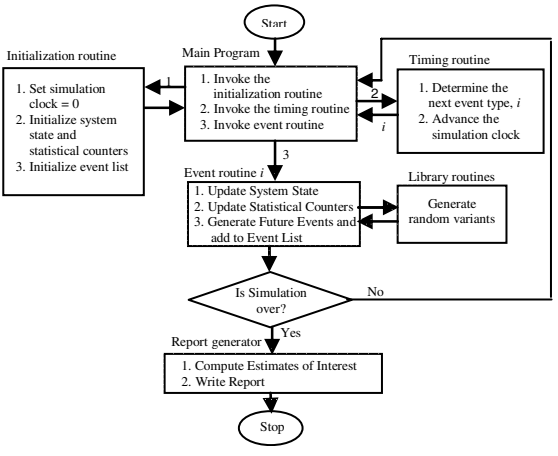


Fig. 6. Flow control for the next-event time-advance approach

The advantage of using this approach is that periods of inactivity can be skipped over by jumping the clock from event time to the next event time. This is perfectly safe; since by definition all state changes only occur at event times. Therefore causality is guaranteed. Furthermore, the event-driven approach is usually exploited in queuing and optimization problems.

7.5 Server Agent

As its name indicates, it is the agent to handle the service process resulting in packet life cycle completion. The two main tasks assigned for this agent are:

Controlling Service Process: as we mentioned earlier, if queue agent detects overload conditions then the Main agent has to change the service policy of the server agent. Hence, the server agent will control the service process to achieve system stability. To perform this task, the server agent follows the steps identified in Fig. 7. The algorithm is based on the idea that the server always serves the packets in the EDF queue (high

priority), and serves the FCFS queue (low priority) if and only if the EDF queue is empty or the FCFS queue reaches its threshold value.

Performing Packet Service: in this task, the server agent has to initially compute the packet queuing time as in the following formula:

$$\text{PacketQueueingTime} = \text{SimclockCurrentTime} - \text{PacketBornTime} \quad (1)$$

After getting this value, a comparison operation is performed to determine whether the packet can proceed in its service process or it is considered to miss its QoS constrains. The comparison operation is defined as in the following:

$$\text{If (PacketQueueingTime} >= \text{PacketQoS)} \rightarrow \text{Missed Packet}$$
$$\text{Otherwise} \rightarrow \text{Proceed Service process} \quad (2)$$

If this operation returns a missed packet, then the server agent will check the flow to which this packet belongs and encounter a miss in that flow. The result will be transferred to the Main agent for the purpose of generating the report at the end of simulation. On the other hand, if the operation results in proceeding the service then the server agent will set its status to busy, request the needed service time from the packet agent, and start the service process. During each packet service, the server agent will advance the clock agent with the packet service time value, increase the number of served packets, and compute the server busy time to be used in calculating the server utilization, by the main agent, at the end of the simulation. The whole service process carried out by the server agent can be shown in Fig. 8.

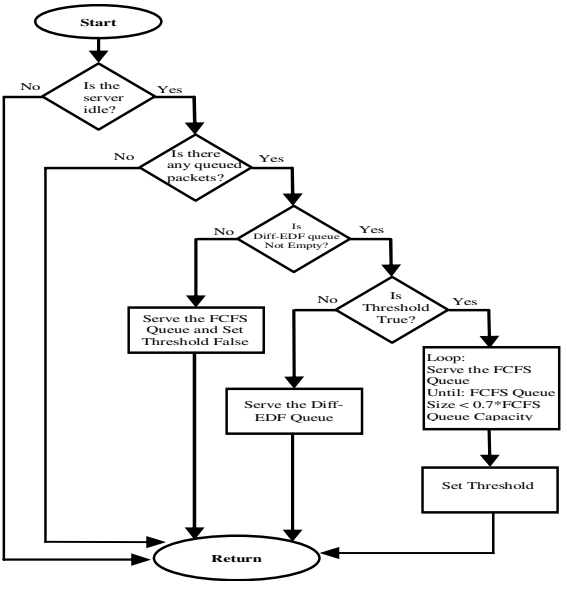


Fig. 7. Service Process Control

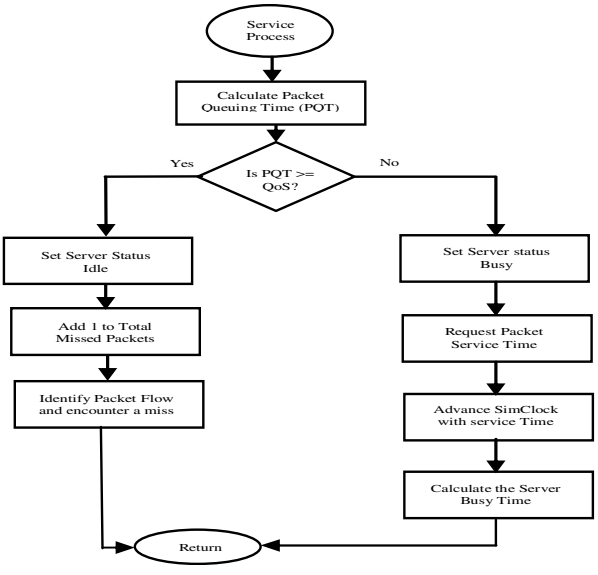


Fig. 8. Packet Service process

7.6 Source Agent

This agent is the system actor of generating the requested traffic determined by the Main agent. It uses several arrival processes' models and the distributed random variants to generate the required traffic. Packet generation steps carried out by the source agent includes determining the exact source which this packet belongs, determining the Packet type, generating packet's inter-arrival time and service time, generating packet deadline, and obtaining initial seeds.

For each packet to be generated there should be an exact time at which the packet born and an exact time at which the packet completes. Consequently, the source agent uses the Exponential Distribution with a mean of $1/\lambda_j$ to generate the packet inter-arrival time, and $1/\mu_j$ to generate the packet service time. For QoS constrains, each packet is associated with a deadline that precisely identify its living period. If the packet exceeds its predefined deadline before being served then it is considered to be useless. Hence, the source agent has to generate this value based on the uniform distribution as in the following formula:

$$\text{PacketQoS} = \text{QoSGenerator.uniform (QoSMax , QoSMin)} \quad (3)$$

Another main task to be handled by the source agent is to ensure that all generated packets obtain an initial seed based on their flow characteristics. This task is carried out by implementing Seed Initializer technique that initializes different seeds for the variety flows as shown below.

```
Private static void assignSeeds(){
For(int StreamID=1,StreamID<=numberOfStreams;StreamID++){
      seed[0]=1973272912;
      double z = 1973272912.0;
      for(int i=1;i<=numberOfStreams;i++){
      do{
            z=MathIEEEremainder(715.0*z,2147483647.0);
            z=MathIEEEremainder(1058.0*z,2147483647.0);
            z=MathIEEEremainder(1385.0*z,2147483647.0);
      }while(z<0.0);
      Seed[i]= (long)z;}}}
```

7.7 Packet Agent

This is a passive agent since it does not have any learning capabilities. It only holds the packet information, received by the source agent, and responds to other agents' requests with the needed information. The packet agent identifies the packet QoS constrains and its type.

8 Agents Management

For our model management, agent actions are reported to the main agent. The main agent receives these reports as agent events. Possible events are Arrival, Departure, Migration, and Termination. Dependent on the agent, any other events are possible.

We discriminate synchronous events where the event emitting agent waits for the main agent acknowledge and asynchronous events where the event emitting agent continues its work without acknowledge. Synchronous events will be used for agent control and monitoring. Asynchronous events represent simple monitoring information. Events like Migration may be implemented as synchronous events. Control of the agent's moves is necessary if domain administrators want to restrict access to certain agent classes. Events like Termination may be asynchronous since it is sufficient to simply know that this particular agent ceases. Keep in mind that event classification is still a matter of research and highly application dependent.

The main agent handles the list of agents. This list comprises all agents that are currently in its domain. Every agent is listed with its identity, class, current location and task identifier. The main agent stores the incoming events and controls, in case of synchronous events, continuation of execution for every agent in the list. An event consists of the event type, the source (i.e. the emitting agent), and some event-specific information such as deadline.

9 Conclusions and Future Work

During the past decade, there has been considerable recent interest in agent-based systems and multi-agent simulation. In this paper, we have presented an adaptive approach to schedule real-time network traffic using the agent based simulation concept. During the multi-agent simulation process, we identified our simulation

stages by decomposing the system into a number of elements, modeling each element with agent, and last describing the possible actions between these agents. Also, we introduced our three-layered ARTAS architecture, agent model, system agents' implementation and management. Consequently, the approach itself had been implemented, for the case of the Hierarchical Diff-EDF real-time scheduler, in the work to be appeared in [19].

As a future work, we suggest to deploy the ARTAS architecture and agent model, proposed in this research, in variety real-time schedulers such as EDF and GPS to experience its efficiency in refining system's design and management.

References

1. Abeck, S., Koppel, A., Seitz, J.: A Management Architecture for Multi-Agent Systems. In: Proceedings of the 3rd IEEE International Workshop on Systems Management, pp. 133–138. IEEE Computer Society Press, Los Alamitos (1998)
2. DiPippo, L.C., Wolfe, V.F.: A Real-Time Multi-Agent System Architecture for E-Commerce Applications. In: Proceedings of the 5th IEEE International Symposium on Autonomous Decentralized Systems, pp. 357–364. IEEE Computer Society Press, Los Alamitos (2001)
3. Drogoul, A., Vanbergue, D., Meuisse, T.: Multi-Agent Based Simulation: Where are the Agents? In: The 3rd International Workshop on Multi-Agent Based Simulation. LNCS, vol. 258, pp. 1–15. Springer, Heidelberg (2002)
4. Gelenbe, E., Seref, E., Xu, Z.: Simulation with Learning Agents. Proceedings of the IEEE. 89(2), 148–157 (2001)
5. Uhrmacher, A.M., Gugler, K.: Distributed, Parallel Simulation of Multiple, Deliberative Agents. In: Proceedings of the 14th IEEE Workshop on Parallel and Distributed Simulation, pp. 101–108. IEEE Computer Society Press, Los Alamitos (2000)
6. Wooldridge, M., Jennings, N.R.: Intelligent Agents: Theory and Practice. Knowledge Engineering Review 10, 115–152 (1995)
7. Bradshaw, J.: Software Agents. AAAI Press, Menlo Park, CA (1997)
8. Jennings, N.R., Wooldridge, M.: Applications of Intelligent Agents, pp. 3–28. Springer, New York (1998)
9. Logan, B., Theodoropoulos, G.: The Distributed Simulation of MultiagentSystems. Proceedings of The IEEE. 89(2), 174–185 (2001)
10. Srinivasan, A.: Efficient and Flexible Fair Scheduling of Real-time Tasks on Multiprocessors. Chapel Hill (2003)
11. Zhang, H., Ferrari, D.: Rate-Controlled Service Disciplines. Journal of High Speed Networks. 3(4), 389–412 (1994)
12. Lu, C., Stankovic, J.A., Tao, G., Son, S.H.: Feedback Control Real-time Scheduling: Framework, Modeling and Algorithms. Special issue of Real-Time Systems Journal on Control-Theoretic Approaches to Real-Time Computing 23(1/2), 85–126 (2002)
13. Brooks, R.A.: A Robust Layered Control System for a Mobile Agent. The IEEE Journal of Robotics Automation Management 2(1), 14–23 (1986)
14. Ferguson, I.A.: Toward An Architecture for Adaptive, Rational, Mobile Agents. In: Proceedings of the Third European Workshop on Modeling Autonomous Agents and Multi-agent Worlds, pp. 249–262 (1992)
15. Kaelbling, L.P.: An Architecture for Intelligent Reactive Systems, pp. 713–728. San Mateo, Morgan Kaufmann (1990)

16. Muller, J.P., Pischel, M.: An Architecture for Dynamically Interacting Agents. International Journal Int. Coop. Information Systems. 3(1), 25–45 (1994)
17. Booch, G.: Object-Oriented Design with Applications. Benjamin/ Cummings, Redwood City, CA (1994)
18. Sarjoughian, H.S., Zeigler, B.P., Hall, S.B.: A Layered Modeling and Simulation Architecture for Agent-Based System Development. Proceedings of the IEEE. 89(2), 201–213 (2001)
19. Saleh, M., Othman, Z.A.: Hierarchical Diff-EDF: An Agent Based Scheduler for Heterogeneous Real-Time Packet Networks. To be appeared in the proceedings of the 6th EUROSIM Congress (2007)

Defining Security Architectural Patterns Based on Viewpoints

David G. Rosado[1], Carlos Gutiérrez[2], Eduardo Fernández-Medina[1], and Mario Piattini[1]

[1] ALARCOS Research Group. Information Systems and Technologies Department UCLM-Indra. Research and Development Institute. University of Castilla-La Mancha Paseo de la Universidad, 4 – 13071 Ciudad Real, Spain
{David.GRosado, Eduardo.Fdez-Medina, Mario.Piattini}@uclm.es
[2] Correos Telecom, Conde de Peñalver, 19 bis 6ª pl. 28006 Madrid, Spain
carlos.gutierrez@correos.es

Abstract. Recently, there has been a growing interest in identifying security patterns in software-intensive systems since they provide techniques for considering, detecting and solving security issues from the beginning of its development life-cycle. This paper describes how security architectural patterns lack of a comprehensive and complete well-structured documentation that conveys essential information of its logical structure, run-time behaviour, deployment-time and monitoring configuration, and so on. Thus we propose a set of security viewpoints to describe software-intensive security patterns adhered to ANSI/IEEE 1471-2000. In order to maximize comprehensibility, we make use of well-known language notations such as UML to represent all the necessary information for defining a software-intensive architectural security pattern conforming to the IEEE 1471-2000 standard. We investigate security architectural patterns from several IEEE 1471-2000 compliant viewpoints.

Keywords: Software Architecture**,** Security patterns, viewpoints, security.

1 Introduction

In most organizations, the importance of application-level security is often underestimated until an application faces a major security breach that causes a serious loss or downtime. Most of the time, it is clear that the probable cause of the failure is related to deficiencies in the application architecture and design, the programming, the coding security, the runtime platform, and the tools and utilities used (e.g.: COTS). The primary responsibility, of course, belongs to the application architects and developers who contributed to the application design and program code. As a result, today it is mandatory to adopt a proactive security approach during the application development life cycle that identifies critical security aspects. Architects and developers today must design security into their applications from the beginning of its lifecycle [1].

Software architecture has emerged as an important sub-discipline of software engineering, particularly in the realm of large system development. Architecture gives

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, pp. 262 – 272, 2007.
© Springer-Verlag Berlin Heidelberg 2007

us intellectual control over a complex system by allowing us to focus on the essential components and their interactions, rather than on extraneous details [2].

The properties that the system exhibits as it executes are among the most important issues to consider when designing, understanding, or implementing a system's architecture. What the system computes is, of course, one of these issues. But nearly as important are properties (i.e., quality attributes) such as performance, reliability, security, or modifiability. The architecture must be documented to communicate how it achieves those properties [2].

Recently, there has been a growing interest in identifying security patterns in software-intensive systems since they provide techniques for considering, detecting and solving security issues from the beginning of its development life-cycle [3][4][5][6]. Security patterns work together to form a collection of coordinated security countermeasures thereby addressing host, network and application security.

This paper describes how security architectural patterns lack of a comprehensive and complete well-structured documentation that conveys essential information of its logical structure, run-time behaviour, deployment-time and monitoring configuration, constraints, elements, and so on. In consequence, we show an alternative way for describing architectures from viewpoints and views, and therefore we can add more information about the pattern in the template used for defining patterns. Therefore we propose a set of viewpoints to define software-intensive security patterns adhered to

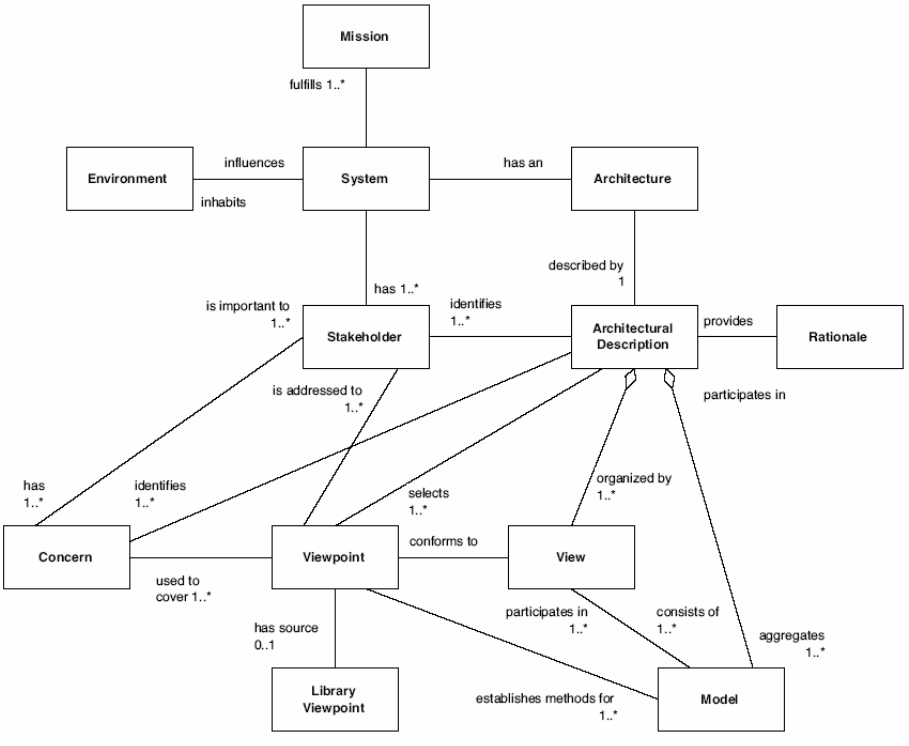


Fig. 1. Conceptual model of architectural description [7]

ANSI/IEEE 1471-2000 [7], the Recommended Practice for Architectural Description. This standard represents an emerging consensus for specifying the content of an architectural description for a software-intensive system. This approach is based on the well-known architectural concept of views [8], and holds that documentation consists of defining the relevant security views and then describing the information that applies to more than one security view. Each view conforms to a viewpoint, which in turn is a realization of the concerns of one or more stakeholders.

To put views and viewpoints in context (that we will define later in the section 2), consider the conceptual model in Fig. 1, which illustrates how views and viewpoints relate to the other important architectural concepts [9].

The remainder of this paper is organized as follows. Section 2 discusses of the importance of defining software architectures and the two most important concept associated with software architecture definition: the view and viewpoint; In section 3, we will define security patterns and what security architectural patterns are; In section 4 we will introduce the viewpoint's model to defining security patterns and we will describe the viewpoint template defined by IEEE 1471-2000 standard; In section 5 an overview of the IEEE 1471-2000 compliant Security Subsystem Design viewpoint's template definition will be shown. Finally, we will put forward our conclusions and future work.

2 Software Architecture Documentation

The architecture must be documented to communicate how it achieves the properties such as performance, reliability, security, or modifiability. Fundamentally, architecture documentation can serve three different functions [2]: a) A means of education. Typically, this means introducing people to the system. The people may be new members of the team, external analysts, or even a new architect; b) A vehicle for communication among stakeholders. A stakeholder is someone who has a vested interest in the architecture. The documentation's use as a communication vehicle will vary according to which stakeholders are communicating; c) A basis for system analysis. To support analysis, the documentation must provide the appropriate information for the particular activity being performed.

Architecture documentation must balance these varied purposes. It should be abstract enough to be quickly understood by new developers. It should be sufficiently detailed so that it serves as a blueprint for its construction. At the same time, it should have enough information so that it can serve as a basis for analysis [2].

Perhaps the most important concept associated with software architecture documentation is the view. A software architecture is a complex entity that cannot be described in a simple one-dimensional fashion.

IEEE 1471 [7] defines Architectural View as a representation of a particular system or part of a system from a particular perspective, and defines Architectural Viewpoint as a template that describes how to create and use an architectural view. A viewpoint includes a name, stakeholders, concerns addressed by the viewpoint, and the modeling and analytic conventions.

3 Security Patterns

Security patterns provide techniques for identifying and solving security issues. They work together to form a collection of best practices (to support a security strategy) and they address host, network and application security [10]. The benefits of using patterns are: they can be revisited and implemented at anytime to improve the design of a system; less experienced and non-expert security practitioners can benefit from the experience of those more fluent in security patterns; they provide a common language for discussion, testing and development; they can be easily searched, categorized and refactored; they provide reusable, repeatable and documented security practices; they do not define coding styles, programming languages or vendors [10].

Several authors (as Kienzle and Elder [11]) identify two broad categories of security patterns: i) Structural patterns that can be implemented in the final product. They encompass design patterns, such as those used by the Gang of Four. They often include diagrams of structure and descriptions of interaction; ii) Procedural patterns that can be used to improve the process for development of security-critical software. They often impact the organization or management of a development project.

Design strategies determine which application tactics or design patterns should be used for particular application security scenarios and constraints [1]. Security patterns are an abstraction of security issues (threats, attacks and vulnerabilities [12]) that address a variety of security requirements and specify the most suitable countermeasures. They can be architectural patterns that depict how a security problem can be resolved architecturally, or they can be defensive design strategies upon which secure code can be later built [1].

An architectural pattern expresses a fundamental structural organization schema for software systems. It provides a set of predefined subsystems, specifies their responsibilities, and includes rules and guidelines for organizing the relationships between them [13]. An architectural pattern is a high-level abstraction. The choice of the architectural pattern to be used is a fundamental design decision in the development of a software system. It determines the system-wide structure and constrains the design choices available for the various subsystems. It is, in general, independent of the implementation language to be used.

There is no specific level of detail for security patterns. Different potential stakeholders of security patterns work at different levels, see different characteristics, functionalities, connections and behavior, and possess different concerns of a same pattern. Current work on defining and describing security patterns [1][5][6][14][15] do not consider all of these levels of detail. Then, we propose a set of security viewpoints, adhered to IEEE 1471-2000 standard, for defining security architectural patterns in such a way that all of the aforementioned stakeholders' issues are addressed.

4 Defining Security Viewpoints for Security Architectural Patterns

A software pattern can be described through a set of properties (a template) such as name, problem, solution, and so on. Templates can be defined as we like but always maintaining the main categories.

Thus, each author can describe all sections he/she considers important according to his/her viewpoint [3]. Some authors [16][17] describe a template for patterns indicating the main categories and characteristics that they consider more important. There are many definitions of patterns following these templates as we can see in [18], where too it makes a comparison between security patterns.

We attempt do more extensive the template adding new information from the stakeholders' viewpoint following as reference the "4+1" view model [8], Fig. 2, where five software architecture's views (Logical, Process, Deployment, Implementation and Use-Case Views) are described.

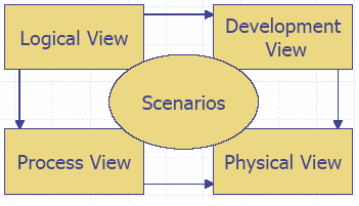


Fig. 2. "4+1" View Model

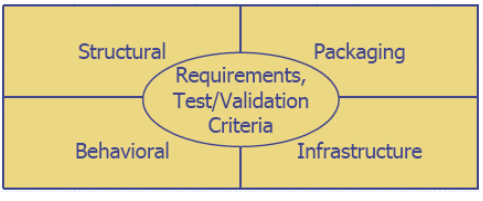


Fig. 3. Newer "4+1" View Model

Actually, the view model is changing (Fig. 3). The logical view and development view are combined into the structural view. The structural view combines both the abstract logical view and the more detailed development view. The process view has been incorporated into the behavioral view. The process view only defines units of execution where the behavioral view includes these plus important behavioral interactions between the architectural elements. The packaging view is new, as component oriented languages allow grouping of structural elements into packages. The new infrastructure view maps closely with the physical view in the older model. The Scenario view in the old model, which represented requirements now includes a logical rendering of them (typically rendered as use-cases), as well as definition of test cases needed to verify the software product.

Obviously, since the 4+1 views preceded IEEE 1471, they do not meet the definition of views as specified in the standard. The 4+1 views are more closely aligned with the concept of viewpoint as defined by IEEE 1471 standard. The 4+1 views describe a collection of representations that provide guidance for software architects. The viewpoints we discuss here are within the spirit of the 4+1 views.

ANSI/IEEE 1471-2000 [7] provides guidance for choosing the best set of views to document, by bringing stakeholder interests to bear. It prescribes defining a set of viewpoints to satisfy the stakeholder community. A viewpoint identifies the set of concerns to be addressed, and identifies the modeling techniques, evaluation techniques, consistency checking techniques, etc., used by any conforming view. A view, then, is a viewpoint applied to a system. It is a representation of a set of software elements, their properties, and the relationships among them that conform to a defining viewpoint. Together, the chosen set of views show the entire architecture and all of its relevant properties. For defining viewpoints, IEEE 1471standard defines a set of elements or sections (template) [19] that can be seen in Fig. 4.

❶ **Abstract**: this section provides a brief overview of the viewpoint;

❷ **Stakeholders and Their Concerns Addressed**: this section describes the stakeholders and their concerns that this viewpoint is intended to address;

❸ **Elements, Relations, Properties, and Constraints**: this defines the types of elements, the relations among them, the significant properties they exhibit, and the constraints they obey for views conforming to this viewpoint;

❹ **Language(s) to Model/Represent Conforming Views**: this section lists the language or languages that will be used to model or represent views conforming to this viewpoint, and cite a definition document for each;

❺ **Applicable Evaluation/Analysis Techniques and Consistency/Completeness Criteria**: this section describes rules for consistency and completeness that apply to views in this viewpoint, as well as any analysis of evaluation techniques that apply to the view that can be used to predict qualities of the system whose architecture is being specified;

❻ **Viewpoint Source**: This section provides a citation for the source of this viewpoint definition, if any.

Fig. 4. Viewpoint Template adhered to IEEE 1471-2000

4.1 Viewpoints Catalogue

We are defining a library of security viewpoints that facilitates and formalizes the defining of security architectural patterns according to IEEE 1471-2000. By definition, these viewpoints are reusable for any software system, thus we can document security patterns, security architecture, software architecture, etc., based on our viewpoint's library.

A number of catalogues of viewpoints already exists, but we have found that all of them do not have security aspects and they are only applied to the development of functional requirements not being considered in the context of the security. In response, we have developed a set of viewpoints for the security architect and the security engineers, that build up and extend the "4+1" set, identified by Philippe Kruchten [8] and Nick Rozanski and Woods [9]. Our catalogue contains seven core security viewpoints: Logical, Process, Development, Physical, Deployment, Operational and Misuse Cases views. Our security viewpoints are represented in Fig. 5.

The *security logical viewpoint* describes the objects or object models within the security architecture that support security behavioral requirements. The *security process viewpoint* describes the security architecture as a logical network of secure communicating processes. This viewpoint assigns each method of the object model to a thread of execution and captures concurrency and synchronization aspects of the security design. The *security physical viewpoint* maps software onto hardware and network elements and reflects the distributed aspect of the security architecture. The *security development viewpoint* focuses on the static organization of the software in the security development environment and deals with issues of configuration

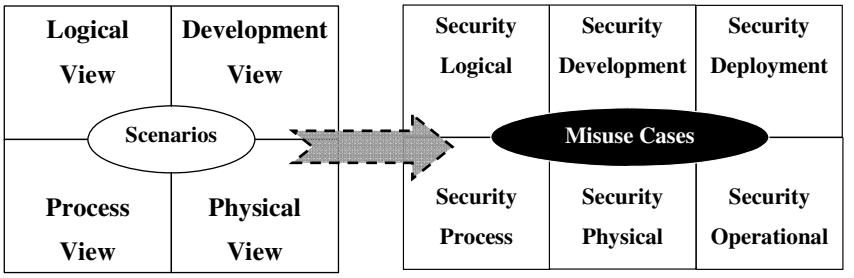


Fig. 5. From "4+1" View Model to Our Security Viewpoints

management, security development assignments, security responsibilities, and countermeasures. The *security deployment viewpoint d*escribes the security environment into which the system will be deployed, including capturing the dependencies the system has on its runtime environment. The aim of the *Security Operational viewpoint* is to identify security system-wide strategies for addressing the operational concerns of the system's stakeholders and to identify solutions that address these.

Moreover, we are defining a new viewpoint's template extending the template of IEEE 1471-2000 aforementioned and we have added new sections in the context of the security as are the follows:

- Security properties to be addressed by the security policy on the basis of the security viewpoint's elements. We consider that the complete security policy of a security pattern is the aggregation of the security policies defined for each security viewpoint.
- Security metrics to be taken into account in this viewpoint.
- Security procedures to be into taken into account from this viewpoint; for instance, from physical viewpoint, procedures to restore the physical node in which the security services defined by the pattern are running, or from logical viewpoint, how to carry out the off-line exchange of key material between the involved parties.
- Best practices: for example from developer's viewpoint, techniques for secure programming, or from physical viewpoint, topologies of secure networks.

5 Definition of Security Design Subsystem Viewpoint

Each viewpoint aforementioned can be divided in different viewpoint satisfying the interest of a particular stakeholder. A series of viewpoints is then used to elaborate the details of the general viewpoint. Selecting the security design subsystem viewpoint and considering the template aforementioned, we defined this viewpoint as presented in Fig. 6. We can say that this viewpoint locates into of the Security Development viewpoint and helps to define it.

❶ **Abstract.** This viewpoint shows security module decomposition and the use between systems of software system. Each security module interprets itself as a subsystem to develop; therefore it is an entity in construction time, that it can communicate with others security subsystems for completing its functionality. From here on a security module defined in a contextual view, we can show its decomposition in security subsystems. The decomposition continues until that each module or subsystem of security is allocated to a unique responsible of development or team.

❷ **Stakeholders and their concerns addressed**. Secure applications will be developed by (at least) three different roles:

- Application software developers that focus on the business logic (1),
- Security providers that focus on the design and implementation of reusable frameworks of security logic (2),
- Security engineers that implement the security policy for a particular application and focuse on how the system is implemented from the perspective of security, and how security affects the system properties. It examines the system to establish what information is stored and processed, how valuable it is, what threats exist, and how they can be addressed.
- Project managers, who must define work assignments, form teams, and formulate project plans and budgets and schedules;
- Maintainers, who are tasked with modifying the software elements;
- Testers and integrators who use the modules as their unit of work.

❸ **Elements, Relations, Properties, and Constraints**.

- Security modules are units of implementation, and its decomposition in shorter modules, just as use dependency existent between them.
- Relations between security modules can have the semantic associated 'is-part-of' or 'utilize'.
- The last level of subsystems called security design subsystems, defined in the views according to this viewpoint must:
  - To be set of products of work of design assigned to different develop teams;
  - Security subsystems will correlate with the construction directories that will be developed, tested and handed over respective teams of development;
  - Following modality origins, the security subsystems must exhibit high cohesion and low coupling;
- These subsystems will be the lower level entities for which the software architects team will need to define the interface;
- Multiple subsystems of security can be assigned to one same development team, but each security subsystem will be developed, tested and versioning of independent form;
- Each security subsystem can be considered as a system to design by the security design team to who have been assigned.

❹ **Language(s) to Model/Represent Conforming Views**.

- The representation language used will be UML and extensions for security aspects as UMLSec [20][21] and of SecureUML [22].
- Each module or subsystem of security will represent itself as a stereotyped UML packet with the reserved word <<subsystem>>. The use relations will show as relations of dependence UML including the stereotype <<uses>> and decomposition relations with nesting of UML packets.

Fig. 6. Security Design Subsystem Viewpoint

- The interfaces that implements each system are modeled as UML interfaces and the name of the service to include in each interface correspond with the names of the use cases defined in the abstraction level of "Goal Summarize" [23] for each subsystem.
- The design subsystem included into views according to this viewpoint will declare a realization of a or more interfaces whose methods correspond with use cases in the abstraction level "User Goal" specified in the model of use cases of this design subsystem.

❺ **Applicable Evaluation/Analysis Techniques and Consistency/Completeness Criteria**. Revision checking with the different development groups of form that they understand the context of the subsystem that they are going to develop (what system comes from) so as the interfaces with others design subsystems. Some analysis and evaluation methods are described by Ronald Wassermann [3] and Jan Jürjens [24].

❻ **Viewpoint Source**. Viewpoint of Design Subsystem [25].

Fig. 6. *(continued)*

6 Conclusions

It is important to describe or document a software architecture because it serves to introduce people to the system, it serves as a vehicle for communication among stakeholders, and it serves as a basis for system analysis. Moreover, an architecture documented is crucial for understanding its main characteristics, its functionality, its components and connections, its behaviour, and so on.

As an architectural pattern is a micro-architecture, too it will be important to or describe its main characteristics for that stakeholder can use and analyze the pattern at the time of integrating it in the design of the application, or in the design of the whole architecture.

In this paper, we have described an architectural pattern from viewpoints attempting to give a vision wider of its main characteristics, of its design, connections, elements, interfaces, implementation, classes and behavior. We have added news sections to the existing templates, extending the information about the pattern. The enhanced security pattern template presented herein contains additional information, including behavior, constraints, and related security principles, that addresses difficulties inherent to the development of security-critical systems. The adoption of IEEE 1471 and the upcoming release of the UML 2.0, UMLSec [21] and SecureUML [22] should help improve the future practice of security software architecture.

Our intention is define security architectural patterns by means of a views template and a viewpoint template as to recommend ANSI/IEEE 1471-2000 [7], that it provides guidance for choosing the best set of views to document. We have defined a viewpoints' catalogue and we have added and we are adding new elements or sections to the viewpoint template of IEEE 1471-2000 standard. We will create a full template of views for security architectural patterns. Our research concentrates in defining a library of viewpoints adhered to IEEE 1471-2000 which

instance are the views that we can define following the documentation IEEE 1471-2000 [19]. Of this form we could have a library of viewpoints to define security architectural patterns.

Acknowledgements

This research is part of the following projects: DIMENSIONS (PBC-05-012-2) and MISTICO (PBC-06-0082) both partiality supported by FEDER and by the "Consejería de Educación y Ciencia de la Junta de Comunidades de Castilla-La Mancha" (Spain), RETISTRUST (TIN2006-26885-E) and ESFINGE (TIN2006-15175-CO5-05) granted by the "Dirección General de Investigación del Ministerio de Educación y Ciencia" (Spain).

References

1. Steel, C., Nagappan, R., Lai, R.: Core Security Patterns, p. 1088. Prentice Hall, Englewood Cliffs (2005)
2. Bachmann, F., Bass, L., Clements, P., Garlan, D., Ivers, J., Little, R., Nord, R., Stafford, J.: Documenting Software Architectures:Organization of Documentation Package, Software Engineering Institute (2001)
3. Cheng, B.H.C., Konrad, S., Campbell, L.A., Wassermann, R.: Using Security Patterns to Model and Analyze Security Requirements. Monterey Bay, CA, USA, pp. 13–22 (2003)
4. Schumacher, M., Fernandez, E.B., Hybertson, D., Buschmann, F.: Security Patterns, 1st edn., p. 512. John Wiley & Sons, Chichester (2005)
5. Schumacher, M., Roedig, U.: Security Engineering with Patterns. In: 8th Conference on Patterns Lnaguages of Programs, PLoP 2001, Monticello, Illinois, USA (2001)
6. Yoder, J., Barcalow, J.: Architectural Patterns for Enabling Application Security, Monticello, Illinois, USA (1997)
7. IEEE, Recommended Practice for Architectural Description of Software-Intensive Systems (IEEE Std 1471-2000). Institute of Electrical and Electronics Engineers: New York, NY, p. 29 (2000), http://standards.ieee.org/reading/ieee/std_public/description/se/1471-2000_desc.html
8. Kruchten, P.: Architectural Blueprints - The "4+1" View Model of Software Architecture. IEEE Software 12(6), 42–50 (1995)
9. Rozanski, N., Woods, E.i.: Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives, 1st edn., p. 576. Addison Wesley, Reading (2005)
10. Berry, C.A., Carnell, J., Juric, M.B., Kunnumpurath, M.M., Nashi, N., Romanosky, S.: Patterns Applied to Manage Security, in J2EE Design Patterns Applied, Ch. 5 (2002)
11. Kienzle, D.M., Elder, M.C.: Final Technical Report: Security Patterns for web Application Development (2005)
12. Firesmith, D.G.: Commom Concepts Underlying Safety, Security, and Survivability Engineering CMU/SEI-2003-TN-033. SEI (2003)
13. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Pattern-Oriented Software Architecture: A System of Patterns, p. 476. John Wiley & Sons, Chichester (1996)
14. Fernandez, E.B., Pan, R.: A pattern language for security models. In: 8th Conference on Pattern Languages of Programs, PLoP 2001. Allerton Park, Illinois, USA (2001)

15. Security Design Patterns (2001), Available on: http://www.cgisecurity.com/lib/security-DesignPatterns.html
16. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1994)
17. AGCS, AG Communication System. Template Pattern (1996)
18. Rosado, D.G., GutiÃ©rrez, C., Fernandez-Medina, E., Piattini, M.: A Study of Security Architectural Patterns, Vienna, Austria, pp. 358–365. IEEE Computer Society, Los Alamitos (2006)
19. Software Architecture Document (SAD) (2006) Available on: www.sei.cmu.edu/architecture/SAD_template2.dot
20. Jurjens, J.: Towards Secure Systems Development with UMLsec. In: Hussmann, H. (ed.) ETAPS 2001 and FASE 2001. LNCS, vol. 2029, pp. 187–200. Springer, Heidelberg (2001)
21. Jurjens, J.: UMLsec: Extending UML for Secure Systems Development. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 412–425. Springer, Heidelberg (2002)
22. Lodderstedt, T., Basin, D., Doser, J.: SecureUML: A UML-Based Modeling Language for Model-Driven Security. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 426–441. Springer, Heidelberg (2002)
23. CockBurn, A.: Writing Effective Use Cases, p. 270. Addison-Wesley Professional, Reading (2000)
24. Deubler, M., Grünbauer, J., Jürjens, J., Wimmel, G.: Sound Development of Secure Service-based Systems. In: Second International Conference on Service Oriented Computing (ICSOC), ACM Press, New York (2004)
25. Garlan, J., Anthony, R.: Large-Scale Software Architecture, p. 278. John Wiley & Sons, Chichester (2002)

A New Nonrepudiable Threshold Proxy Signature Scheme with Valid Delegation Period

Min-Shiang Hwang[1], Shiang-Feng Tzeng[2], and Chun-Ta Li[3]

[1] Department of Management Information Systems, National Chung Hsing University, 250 Kuo Kuang Road, 402 Taichung, Taiwan, R.O.C.
mshwang@nchu.edu.tw
[2] Department of Computer Science and Information Engineering, National Central University, Taiwan, R.O.C.
[3] Department of Computer Science, National Chung Hsing University, Taiwan, R.O.C.

Abstract. In the (t, n) proxy signature scheme, the signature, signed by a signer originally, can be signed by t or more proxy signers out of a proxy group of n proxy signers. In this paper, we propose a new secure non-repudiable threshold proxy signature scheme with valid delegation period. In the scheme, the actual signers from the proxy group are known and can be identified. Furthermore, the proposed scheme is efficient in terms of computational complexity.

1 Introduction

The concept of the proxy signature scheme was first introduced by Mambo et al. [18] in 1996. A proxy signature scheme allows the original signer to delegate the signing capability to a designated person, called a proxy signer, to sign on behalf of the original signer in a way that the following requirements are satisfied [9, 10]:

1. Unforgeability: On behalf of the original signer, a designated proxy signer can create a valid proxy signature. Any third party who is not designated as a proxy signer cannot create a valid proxy signature of the proxy signer.
2. Verifiability: A verifier can be convinced by the original signer's agreement on the signed message from proxy signatures.

So far, proxy signature schemes have been widely discussed [1–3, 11–13, 15–17, 20–24]. There have been three types of delegation: full delegation, partial delegation, and delegation by the warrant [18]. In full delegation, a proxy signer is given the same secret key that the original signer has. Hence, the signature of the proxy signer is indistinguishable from the one created by the original signer. In partial delegation, the original signer uses the original secret key to create a proxy signature key which will be sent to the proxy signer. For security requirements, it is computationally infeasible for the proxy signer to derive the original signer's secret key from the given proxy signature key. The proxy signer

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 273–284, 2007.
© Springer-Verlag Berlin Heidelberg 2007

can use the proxy signature key to sign messages on behalf of the original signer. Finally, the third type of delegation uses a warrant to certify that the proxy signer is delegated by the original signer.

According to the above description, partial delegation and delegation by the warrant are more secure than full delegation. Compared with delegation by the warrant, partial delegation needs less processing time. Therefore, among these three types of delegation, partial delegation seems to be the best choice. In this paper, we shall focus on the proxy signature authorized with partial delegation.

A digital signature scheme allows the signers to sign messages in such a way that everyone can verify the validity of the authentic signatures [5]. However, nobody can forge the signatures of other messages. Hence the signer cannot repudiate the signatures that she/he has ever signed. This property is usually referred to as "non-repudiation". However, in the proxy signature scheme, there exists a dispute as to partial delegation where, aside from the proxy signature key for the proxy signer, the original signer can derive another proxy signature key and save it for her/his own use to generate a valid proxy signature and impersonate the proxy signer. This problem can be solved using the non-repudiation property, the capability of identifying the actual signer, to further strengthen the proxy signature scheme.

Following the development of the proxy signature scheme, some threshold proxy signature schemes were widely studied [14, 24]. In a (t, n) threshold proxy signature scheme, which is a variant of the proxy signature scheme, the proxy signature key is shared among a group of n proxy signers delegated by the original signer. Any t or more proxy signers can cooperatively sign messages on behalf of the original signer. So far, there have been at least two threshold proxy signature schemes proposed [14, 24]. Kim's scheme [14] is non-repudiable, but Zhang's scheme [24] is not.

Based on Kim's scheme, Sun [19] proposed an efficient non-repudiable threshold proxy signature scheme with a known signer. Sun's scheme is more efficient than the other threshold proxy signature schemes. The main advantage of Sun's scheme is that the verifier is able to identify the actual signers in the proxy group. However, the weakness of Sun's scheme is that it is vulnerable to the conspiracy attack [6, 8]. Hsu et al. [6] proposed a new non-repudiable proxy signature scheme with known signers that can withstand the conspiracy attack and is more efficient than Sun's scheme. Technically, the security of the above two non-repudiable threshold proxy signature schemes is based on the discrete logarithm (DL) problem [5, 7].

In this paper, a new secure non-repudiable proxy signature scheme will be proposed where the verifier is able to identify the actual signers in the proxy group. Besides, the new scheme will be able to withstand the conspiracy attack. Moreover, the proposed scheme divides the valid delegation period of proxy signature into many pieces. In the different pieces of valid delegation period exists a different proxy signature key. The goal is to increase security.

In this article, we shall first briefly review Hsu-Wu-Wu scheme in the following section. Then, in Section 3, our new secure non-repudiable proxy signature

scheme will be proposed. The security and the performance analysis of the proposed scheme will be separately discussed in Sections 4 and 5. Finally, the concluding remarks will be in the last section.

2 Review of Hsu-Wu-Wu Scheme

In this section, we will overview Hsu-Wu-Wu scheme [6]. Their scheme can be divided into four phases: secret share generation, proxy share generation, proxy signature generation, and proxy signature verification. There exists a system authority (SA) whose tasks are to initialize the system and to manage the public directory. In the secret share generation phase, SA is the party that takes all the action. Initially, SA first selects and publishes the following parameters: p denotes a large prime; q denotes a large prime factor of $p-1$; g denotes a generator in $GF(p)$ with order q; $h(\cdot)$ denotes a one-way hash function; m_w denotes a warrant, which records the identity of the original signer, the proxy group of the proxy signers, the parameters t and n, and the valid delegation time, etc; and $ASID$ denotes an identity of the actual signer.

Each user P_i, having the public identity $v_i \in Z_q$, now owns a secret key $x_i \in Z_q^*$ and a public key $y_i = g^{x_i} \bmod p$ which is certified by the certificate authority (CA). Let P_0 be the original signer and $G = \{P_1, P_2, \cdots, P_n\}$ be the proxy group of n proxy signers.

2.1 Secret Share Generation Phase

SA first chooses the group secret key X_G and calculates the group public key $Y_G = g^{X_G} \bmod p$ which is certified by CA. Then, SA randomly generates a $(t-1)$-degree polynomial as

$$f(v) = X_G + a_1 v + a_2 v^2 + \cdots + a_{t-1} v^{t-1} \bmod q \tag{1}$$

with random integers $a_i \in Z_q$, $i = 1, 2, \cdots, (t-1)$.

For each $P_i \in G$, SA calculates the secret share $\gamma_i = f(v_i)$ and the corresponding public information $\tau_i = g^{\gamma_i} \bmod p$, where v_i is the public identity for P_i. After that, SA separately sends γ_i to P_i via a secure channel and publishes all τ_i's.

2.2 Proxy Share Generation Phase

The original signer P_0 performs the following steps to delegate the signing capability to G.

1. Choose a random integer $k \in Z_q^*$ and compute $K = g^k \bmod p$.
2. Compute the proxy signature key as $\sigma = k + x_0 h(m_w || K) \bmod q$.
3. Choose a polynomial

$$f_0(v) = \sigma + b_1 v + b_2 v^2 + \cdots + b_{t-1} v^{t-1} \bmod q \tag{2}$$

with random integers $b_j \in Z_q$, $j = 1, 2, \cdots, (t-1)$.

4. Publish $B_j = g^{b_j} \bmod p$, for $j = 1, 2, \cdots, (t-1)$.
5. Send σ_i $(= f_0(v_i))$ to $P_i \in G$ via a secure channel.
6. Broadcast (m_w, K) to G.

Upon receiving σ_i, each $P_i \in G$ can validate it by checking the following equation: $g^{\sigma_i} \stackrel{?}{=} y_0^{h(m_w \| K)} K(\prod_{j=1}^{t-1} B_j^{v_i^j}) \bmod p$. If it holds, P_i calculates

$$\sigma_i' = \sigma_i + \gamma_i h(m_w || K) \bmod q \tag{3}$$

as her/his proxy share.

2.3 Proxy Signature Generation Phase

Given a message m, any t or more proxy signers of G will be proxies for P_0 to sign m in this phase. Without loss of generality, let $D = \{P_1, P_2, \cdots, P_t\}$ be the actual proxy signers and $ASID$ (Actual Signers' ID) be the collection of the identities of all the users in D. They cooperatively sign m on behalf of P_0 by performing the following steps.

1. Each $P_i \in D$ chooses a random integer $k_i \in Z_q^*$ and then broadcasts r_i $(= g^{k_i} \bmod p)$.
2. Upon obtaining all r_j's, $j = 1, 2, \cdots, t$, each $P_i \in D$ calculates

$$R = \prod_{j=1}^{t} r_j \bmod p,$$
$$s_i = k_i R + (L_i \sigma_i' + x_i) h(R || ASID || m) \bmod q,$$

   where $L_i = \prod_{j=1, j \neq i}^{t} (-v_j)(v_i - v_j)^{-1} \bmod q$. Here, s_i is the individual proxy signature which is sent to the designated clerk.
3. Upon receiving s_i, the designated clerk validates it by checking

$$g^{s_i} \stackrel{?}{=} r_i^R (((y_0 \tau_i)^{h(m_w || K)} (\prod_{j=1}^{t-1} B_j^{v_i^j}) K)^{L_i} y_i)^{h(R || ASID || m)} \bmod p.$$

   If it holds, (r_i, s_i) is the valid individual proxy signature of m. If all the individual proxy signatures for m are valid, the clerk calculates $S = \sum_{j=1}^{t} s_j \bmod q$.

The proxy signature of m is $(R, S, K, m_w, ASID)$.

2.4 Proxy Signature Verification Phase

When a verifier receives a proxy signature $(R, S, K, m_w, ASID)$ for m, the verifier can identify the original signer and the proxy group with the warrant m_w. Then, she/he knows the actual signers from $ASID$ and obtains the necessary

public keys from CA. Thereafter, the verifier can validate the proxy signature by checking if

$$g^S \stackrel{?}{=} R^R (K(y_0 Y_G)^{h(m_w||K)} \prod_{i=1}^{t} y_i)^{h(R||ASID||m)} \bmod p.$$

If it holds, the proxy signature $(R, S, K, m_w, ASID)$ for m is valid.

3 The Proposed Scheme

In this section, a new secure non-repudiable (t, n) threshold proxy signature scheme with known signers is to be proposed. The scheme has four phases: secret share generation, proxy share generation, proxy signature generation, and proxy signature verification. The system parameters p, q, g, $h(\cdot)$, m_w and $ASID$ are the same as those in Section 2.

The original signer's private key $x_0 \in Z_q^*$ is randomly selected, and the public key is $y_0 = g^{x_0} \bmod p$ certified by a CA. Each proxy signer P_i owns a private key $x_i \in Z_q^*$ and a public key $y_i = g^{x_i} \bmod p$ which is also certified by a CA. Let P_0 be the original signer and $G = \{P_1, P_2, \cdots, P_n\}$ be the proxy group of n proxy signers.

3.1 Secret Share Generation Phase

1. P_0 randomly chooses a secret share s_i, $i = 1, 2, \cdots, n$, for each $P_i \in G$. Then, P_0 delivers s_i to each P_i via a secure channel.
2. P_0 declares the valid delegation period T within which each proxy signer $P_i \in G$ is capable of signing the proxy signature. Then, P_0 divides T into pieces T_u for $u = 1, 2, \cdots, r$. For example, P_0 declares that the valid delegation period T is one day. Suppose one delegation period unit is one hour. Then, P_0 divides T into 24 parts T_u for $u = 1, 2, \cdots, 24$. In other words, when each $P_i \in G$ works as a proxy, P_i must choose the corresponding delegation period T_u accordingly. For instance, let T_1 be $13:00 \sim 14:00$ and T_2 be $14:00 \sim 15:00$. When it is $13:30$, each $P_i \in G$ signs a message m on behalf of P_0. In this case, they should choose delegation period T_1.
3. P_0 uses the secret share s_i of each $P_i \in G$ through a one-way hash function $h(\cdot)$, and T_u generates proxy share s_{ij} for each delegation period T_u, $i = 1, 2, \cdots, n$, $u = 1, 2, \cdots, r$ as

$$s_{iu} = h(s_i, T_u).$$

4. P_0 constructs a $(n-1)$-degree polynomial $f_u(x)$ through each (ID_i, s_{iu}) of P_i from the Lagrange interpolation polynomial [4] for $i = 1, 2, \cdots, n$, $u = 1, 2, \cdots, r$. That is,

$$\begin{aligned} f_u(x) &= \sum_{i=1}^{n} s_{iu} \prod_{j=1, j\neq i}^{n} \frac{x - ID_j}{ID_i - ID_j} \quad (u = 1, 2, ..., r) \\ &= d_u + b_{1,u}x + b_{2,u}x^2 + \cdots + b_{n-1,u}x^{n-1}. \end{aligned}$$

The secret d_u $(= f_u(0))$ is the uth secret to be shared among $P_i \in G$ in T_u.

5. P_0 chooses random integer x_{uk} $(x_{uk} \neq ID_i, i = 1, 2, ..., n)$, and computes y_{uk} $(= f_u(x_{uk}))$ by each $(n-1)$-degree polynomial $f_u(x)$ for $k = 1, 2, ..., (n-t)$.
6. P_0 calculates the public key e_u satisfying $e_u = g^{d_u} \bmod p$ for each secret d_u, $u = 1, 2, \cdots, r$.
7. P_0 attaches the delegation information to the warrant m_w which records the valid delegation period T_u and some corresponding values, the public key e_u and (x_{uk}, y_{uk}) for $u = 1, 2, ..., r, k = 1, 2, ..., (n-t)$.

Finally, P_0 delivers s_i to each $P_i \in G$ via a secure channel.

3.2 Proxy Share Generation Phase

P_0 executes the following steps to delegate the signing capability to G.

1. Choose a random number $t_0 \in Z_q^*$ and compute the parameter k_0.

$$k_0 = g^{t_0} \bmod p$$

2. Compute the proxy signature key as

$$\sigma = t_0 + x_0 h(m_w || k_0) \bmod q.$$

3. Send (σ, m_w, k_0) to $P_i \in G$ via a public channel.

After receiving (σ, m_w, k_0), each $P_i \in G$ checks whether or not the following equation holds,

$$g^{\sigma} \stackrel{?}{=} k_0 y_0^{h(m_w||k_0)} \bmod p.$$

If it holds, each P_i uses σ as his proxy share.

3.3 Proxy Signature Generation Phase

Given a message m, any t or more proxy signers will be proxies for P_0 to sign m in this phase. Without loss of generality, let $D = \{P_1, P_2, \cdots, P_t\}$ be the actual proxy signers. They cooperatively sign m on behalf of P_0 by performing the following steps:

1. Each $P_i \in D$ chooses a valid delegation period T_u and finds some corresponding values, the public key e_u and public parameters (x_{uk}, y_{uk}) from P_0's warrant m_w.
2. Each $P_i \in D$ calculates the secret s_{iu} as

$$s_{iu} = h(s_i, T_u)$$

and sends (ID_i, s_{iu}) to each P_j for $j = 1, 2, ..., t, j \neq i$.

3. Upon receiving (ID_i, s_{iu}), each $P_i \in D$ uses t pairs of (ID_i, s_{iu}) and $(n-t)$ pairs of public parameters (x_{uk}, y_{uk}) to construct the $(n-1)$-degree polynomial $f_u(x)$ from the Lagrange interpolation polynomial as follows:

$$f_u(x) = d_u + b_{1,u}x + b_{2,u}x^2 + \cdots + b_{n-1,u}x^{n-1}.$$

Here, the secret d_u can be derived.

4. Each $P_i \in D$ chooses a random number $t_i \in Z_q^*$ and broadcasts k_i.

$$k_i = g^{t_i} \bmod p$$

5. For each received k_j $(j = 1, 2, ..., t, j \neq i)$, each $P_i \in D$ computes

$$R = \prod_{i=1}^{t} k_i \bmod p,$$
$$s_i = t_i R + ((\sigma + d_u)t^{-1} + x_i)h(R||m||ASID) \bmod q.$$

Here, s_i is the individual proxy signature which is sent to the designated clerk.

6. For each received s_i, the designated clerk checks whether the following equation holds

$$g^{s_i} \stackrel{?}{=} k_i^R((k_0 y_0^{h(m_w||k_0)} e_u)^{t^{-1}} y_i)^{h(R||m||ASID)} \bmod p.$$

If it holds, (r_i, s_i) is a valid individual proxy signature of m. If all individual proxy signatures are valid, the designated clerk computes

$$S = \sum_{i=1}^{n} s_i \bmod q.$$

The proxy signature of m is $(m_w, k_0, R, S, T_u, ASID)$.

3.4 Proxy Signature Verification Phase

On receiving the proxy signature $(m_w, k_0, R, S, T_u, ASID)$ for m, any verifier can verify the validity of the proxy signature by following the steps below:

1. From m_w and $ASID$, the verifier can get the public key y_i of P_0 and $P_i \in G$ from CA and know who the original signer and the actual proxy signers are.
2. The verifier can use T_u of the proxy signature to find the corresponding value, the public key e_u, from m_w.
3. The verifier checks the validity of the proxy signature of the m through the following equation:

$$g^S \stackrel{?}{=} R^R(k_0 y_0^{h(m_w||k_0)} e_u \prod_{i=1}^{t} y_i)^{h(R||m||ASID)} \bmod p.$$

If it holds, the signature $(m_w, k_0, R, S, T_u, ASID)$ for m is a valid proxy signature and is cooperatively generated by the proxy signers $P_1, P_2, \cdots, P_t$ on behalf of the proxy group and the original signer.

4 Security Analysis

The security of the proposed scheme is similar to that of Hsu-Wu-Wu scheme based on the well-known one-way hash function and the discrete logarithm problem [5] cryptographic assumptions. In this section, we shall make a comparison between the proposed scheme and Hsu-Wu-Wu scheme in terms of confidentiality as well as possible compromising and conspiracy attacks. In Hsu-Wu-Wu scheme, any t users in G can work together to separately reconstruct X_G and σ from Equations (1) and (2) with their γ_i's and σ_i's. They can easily derive any other user P_j's γ_j and σ_j. That is, they also can compute P_j's σ'_j by Equation (3). However, it is unreasonable that they can do these.

In the proposed scheme, each $P_i \in G$ has to protect only one secret share s_i. Furthermore, each $P_i \in G$ can compute a different proxy share s_{iu} in T_u, and each delegation period T_u can obtain the corresponding public key e_u. Through the comparison, we conclude that the proposed scheme outperforms Hsu-Wu-Wu scheme in terms of key confidentiality.

The compromising attack is invoked by any attacker who tries to derive the secret keys d_u for $u = 1, 2, \cdots, r$ from the released public information. It is difficult to derive the secret share d_u from the public information e_u of the m_w under the DL cryptographic assumptions.

Now we consider the conspiracy attack when malicious proxy signers from the proxy group attempt to forge a valid proxy signature and claim that it is generated by some specific proxy signers from the same proxy group. In the proposed scheme, any t proxy signers in G can work together to separately reconstruct d_u and the actual proxy signers' secret keys have used to sign the signature s_i. Therefore, it is necessary for a verifier to verify the signature using the actual proxy signers' public keys. Without the knowledge of the actual proxy signers' secret keys, the malicious proxy signers cannot impersonate other proxy signers and generate proxy signatures in the proposed scheme. As a result, the proxy group cannot deny the proxy signatures that they have signed on behalf of the original signer.

5 Performance Evaluation

In this section, we give a comparison between Hsu-Wu-Wu scheme and the proposed scheme in term of computational complexity. Performance evaluation notations are defined as follows:

- T_{exp}*:* The time for a modular exponentiation computation
- T_{mul}*:* The time for a modular multiplication computation
- T_{inv}*:* The time for a modular inverse computation
- T_h*:* The time for computing a one-way hash function $h(\cdot)$

The results of the comparison of computational complexity between the proposed scheme and Hsu-Wu-Wu scheme are listed in Table 1. In Table 1, we can see that the proposed scheme is more efficient than Hsu-Wu-Wu scheme because of its lower computational complexity.

Table 1. The comparison on computational complexity

	Hsu et al.'s scheme	The proposed scheme
Secret share generation	$(n+1)T_{exp} + n(t-1)T_{mul}$	$T_{exp} + (3n^2 - (t+3)n + t)T_{mul} + nT_{inv} + nT_h$
Proxy share generation	The original signer: $tT_{exp} + (nt - n + 1)T_{mul} + T_h$ Each proxy signer: $(t+1)T_{exp} + (2t-1)T_{mul} + T_h$	The original signer: $T_{exp} + T_{mul} + T_h$ Each proxy signer: $2T_{exp} + T_{mul} + T_h$
Proxy signature generation	Individual proxy signature: $T_{exp} + (3t-1)T_{mul} + (t-1)T_{inv} + T_h$ Proxy signature: $(t^2+4t)T_{exp} + (4t^2 - t - 1)T_{mul} + (t^2 - t)T_{inv} + 2T_h$	Individual proxy signature: $T_{exp} + (2n^2 - 2n + t + 2)T_{mul} + (n+1)T_{inv} + 2T_h$ Proxy signature: $(3t+2)T_{exp} + (3t+1)T_{mul} + T_{inv} + 2T_h$
Proxy signature verification	$4T_{exp} + (t+3)T_{mul} + 2T_h$	$4T_{exp} + (t+3)T_{mul} + 2T_h$

Table 1 shows the computational complexity of four phases of the Hsu-Wu-Wu and the proposed schemes in Sections 2 and 3. Suppose the valid delegation period T is a delegation period unit. In secret share generation of the proposed scheme, the Steps 1 and 2 only need to randomly choose secret shares and declare some parameters which are similar to that of Hsu-Wu-Wu scheme. We ignore this computation. Then, it requires nT_h for Step 3, $(2n^2-2n)T_{mul}$ and nT_{inv} for Step 4, $(n^2-(t+1)n+t)T_{mul}$ for Step 5, and T_{exp} for Step 6 in Section 3, where n denotes the proxy group of n proxy signers; t denotes proxy threshold value, i.e., the number of actual proxy signers. The total computational complexity of secret share generation phase requires $T_{exp} + (3n^2 - (t+3)n + t)T_{mul} + nT_{inv} + nT_h$. However, SA of Hsu-Wu-Wu scheme requires $(n+1)T_{exp} + n(t-1)T_{mul}$. It is obvious that the proposed scheme is efficient than Hsu-Wu-Wu scheme in this phase.

In the proxy share generation phase of the proposed scheme, the original signer of the proposed scheme requires T_{exp} for Step 1 and $T_{mul} + T_h$ for Step 2 in Section 3. The total computational complexity of the original signer requires $T_{exp} + T_{mul} + T_h$. Then, each proxy signer requires $2T_{exp} + T_{mul} + T_h$ for Step 3 in Section 3. However, the original signer and each proxy signer of Hsu-Wu-Wu scheme require $tT_{exp} + (nt - n + 1)T_{mul} + T_h$ and $(t+1)T_{exp} + (2t-1)T_{mul} + T_h$, respectively. It is obvious that the proposed scheme is efficient than Hsu-Wu-Wu scheme in this phase.

In proxy signature generation phase of the proposed scheme, each $P_i \in D$ requires T_h for Step 2, $(2n^2 - 2n)T_{mul} + nT_{inv}$ for Step 3, T_{exp} for Step 4 and $(t+2)T_{mul} + T_{inv} + T_h$ for Step 5 in Section 3. The total computational complexity of each $P_i \in D$ requires $T_{exp} + (2n^2 - 2n + t + 2)T_{mul} + (n+1)T_{inv} + 2T_h$.

The designated clerk requires $(3t+2)T_{exp}+(3t+1)T_{mul}+T_{inv}+2T_h$ for Step 6 in Section 3. However, the producing the individual proxy signature and the producing the proxy signature of Hsu-Wu-Wu scheme require $T_{exp}+(3t-1)T_{mul}+(t-1)T_{inv}+T_h$ and $(t^2+4t)T_{exp}+(4t^2-t-1)T_{mul}+(t^2-t)T_{inv}+2T_h$, respectively. The proposed scheme is also efficient than Hsu-Wu-Wu scheme in this phase.

In proxy signature verification phase of the proposed scheme, any verifier requires $4T_{exp}+(t+3)T_{mul}+2T_h$ to verify the valid signature in Section 3. It is similar of the Hsu-Wu-Wu scheme. Through the comparisons, we conclude that the proposed scheme outperforms Hsu-Wu-Wu's scheme in terms of computational complexities.

6 Conclusions

In this paper, we have proposed a new non-repudiable threshold proxy signature scheme. The proposed scheme is based on the discrete logarithm problem and overcomes the security weakness of Hsu-Wu-Wu scheme. The proposed scheme can identify who the actual proxy signers are from the proxy group who have actually signed the message on behalf of the proxy group and hence the original signer. These actual signers cannot deny the fact that they have signed the signatures and they generate the proxy signature using the different proxy signature key in the different valid delegation period. Furthermore, the proposed scheme is more efficient than Hsu-Wu-Wu scheme in terms of the computational complexity.

Acknowledgements

This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grants NSC 95-2218-E-001-001, and NSC95-2218-E-011-015.

References

1. Awasthi, A.K, Lal, S.: ID-based Ring Signature and Proxy Ring Signature Schemes from Bilinear Pairings. International Journal of Network Security 4(2), 187–192 (2007)
2. Cao, T., Mao, X.: Original Signer's Forgery Attacks on Discrete Logarithm Based Proxy Signature Schemes. International Journal of Network Security 4(3), 355–360 (2007)
3. Das, M.L., Saxena, A., Phatak, D.B.: Proxy Signature Scheme with Effective Revocation Using Bilinear Pairings. International Journal of Network Security 4(3), 312–317 (2007)

4. Dorothy, E., Denning, R.: Cryptography and Data Security. Addison-Wesley, Massachusetts (1982)
5. ElGamal, T.: A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory 31, 469–472 (1985)
6. Hsu, C.-L., Wu, T.-S., Wu, T.-C.: New nonrepudiable threshold proxy signature scheme with known signers. The Journal of Systems and Software 58, 119–124 (2001)
7. Hwang, M.-S., Lee, C.-C., Lu, E.J.-L.: Cryptanalysis of the batch verifying multiple DSA-type digital signatures. Pakistan Journal of Applied Sciences 1(3), 287–288 (2001)
8. Hwang, M.-S., Lin, I.-C., Lu, E.J.-L.: A secure nonrepudiable threshold proxy signature scheme with known signers. International Journal of Informatica 11(2), 1–8 (2000)
9. Hwang, M.-S., Lu, E.J.-L., Lin, I.-C.: A practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem. IEEE Transactions on Knowledge and Data Engineering 15(6), 1552–1560 (2003)
10. Hwang, M.-S., Tzeng, S.-F., Tsai, C.-S.: Generalization of proxy signature based on elliptic curves. Computer Standards & Interfaces 26(2), 73–84 (2004)
11. Hwang, M.-S., Chang, T.-Y.: Threshold Signatures: Current Status and Key Issues. International Journal of Network Security 1(3), 123–137 (2005)
12. Lu, E.J.-L., Huang, C.-J.: A Time-stamping Proxy Signature Scheme Using Time-stamping Service. International Journal of Network Security 2(1), 43–51 (2006)
13. Guo, L., Liu, Y.: Security Analysis and Improvement of Hsu et al. Threshold Proxy Signature Scheme. International Journal of Network Security 2(1), 69–72 (2006)
14. Kim, S., Park, S., Won, D.: Proxy signatures, revisited. In: Han, Y., Quing, S. (eds.) ICICS 1997. LNCS, vol. 1334, pp. 223–232. Springer, Heidelberg (1997)
15. Li, J., Wang, S.: New Efficient Proxy Blind Signature Scheme Using Verifiable Self-certified Public Key. International Journal of Network Security 4(2), 193–200 (2007)
16. Li, L.-H., Tzeng, S.-F., Hwang, M.-S.: Generalization of proxy signature based on discrete logarithms. Computers & Security 22(3), 245–255 (2003)
17. Lu, R., Cao, Z., Shao, J.: On Security of Two Nonrepudiable Threshold Multi-proxy Multi-signature Schemes with Shared Verification. International Journal of Network Security 4(3), 248–253 (2007)
18. Mambo, M., Usuda, K., Okamoto, E.: Proxy signatures: Delegation of the power to sign message. IEICE Trans. Fundamentals E79-A, 1338–1353 (1996)
19. Sun, H.M.: An efficient nonrepudiable threshold proxy signature scheme with known signers. Computer Communications 22(8), 717–722 (1999)
20. Tsai, C.-S., Tzeng, S.-F., Hwang, M.-S.: Improved non-repudiable threshold proxy signature scheme with known signers. Informatica 14(3), 393–402 (2003)
21. Tzeng, S.-F., Hwang, M.-S., Yang, C.-Y.: An improvement of nonrepudiable threshold proxy signature scheme with known signers. Computers & Security 23(2), 174–178 (2004)

22. Yang, C.-Y., Tzeng, S.-F., Hwang, M.-S.: On the efficiency of nonrepudiable threshold proxy signature scheme with known signers. The Journal of Systems and Software 73(3), 507–514 (2004)
23. Xie, Q., Yu, X.-Y.: Cryptanalysis of Two Nonrepudiable Threshold Proxy Signature Schemes. International Journal of Network Security 3(1), 18–22 (2006)
24. Zhang, K.: Threshold proxy signature schemes. In: 1997 Information Security Workshop, pp. 191–197 (1997)

Two-Stage Interval Krawczyk-Schwarz Methods with Applications to Nonlinear Parabolic PDE

Hartmut Schwandt

Technische Universität Berlin, Fakultät II, Institut für Mathematik, MA 6-4, D-10623 Berlin, Germany
schwandt@math.tu-berlin.de

Abstract. By using interval techniques, it is possible to obtain global convergence properties and verified enclosures in the numerical solution of several classes of nonlinear systems of equations. In the present paper, we introduce Newton-like interval methods of the so-called Krawczyk-type for systems arizing from discretizations of almost linear parabolic problems. Parallelism is introduced by domain decomposition and an adaptation of the Schwarz Alternating Procedure. Numerical results from a Sun Opteron cluster are included.

Keywords: almost linear parabolic problems, enclosure methods, nonlinear systems of equations, Newton-like methods, Krawczyk operator, Schwarz Alternating Procedure, domain decomposition, parallel methods.

1 Introduction

Interval arithmetic techniques are available for a wide range of applications. The main objective of interval techniques and methods consists in the computation of intervals in order to get enclosures of inexact data or of solutions of (systems) of equations. In the present paper, we derive Newton-like interval methods for the solution of some classes of nonlinear systems of equations. Unlike for non-interval, i.e. so-called point methods, the convergence of interval methods can be controlled by enclosure in each iterative step. In addition, the methods have global convergence properties under appropriate conditions. In contrast to the wellknown noninterval monotically convergent methods [4], no convexity condition is needed. We derive fast and reliable methods for the numerical solution of large nonlinear systems resulting from the application of difference methods to some almost linear parabolic PDE. An algebraic domain decomposition leads to parallel methods which are based on the Schwarz Alternating Procedure and methods introduced in [2].

2 Notation

For basics of interval arithmetic and interval computations, see [1], e.g. In order to profit from enclosure properties, in interval methods we principally compute with real compact intervals $\mathrm{X} = [\underline{\mathrm{X}}, \overline{\mathrm{X}}]$ with lower and upper bound $\underline{\mathrm{X}}, \overline{\mathrm{X}} \in \mathbb{R}$ instead of using real numbers. We define by $d(\mathrm{X}) := \overline{\mathrm{X}} - \underline{\mathrm{X}}$, $m(\mathrm{X}) := \frac{1}{2}(\overline{\mathrm{X}} + \underline{\mathrm{X}})$,

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 285–297, 2007.
© Springer-Verlag Berlin Heidelberg 2007

$|\mathrm{X}| := \max\{|\underline{\mathrm{X}}|, |\overline{\mathrm{X}}|\}$, $q(\mathrm{X},\mathrm{Y}) = \max\{|\underline{\mathrm{X}} - \underline{\mathrm{Y}}|, |\overline{\mathrm{X}} - \overline{\mathrm{Y}}|\}$ the width, midpoint and absolute value, resp. of an interval X and the distance of two intervals X, Y.

Small (capital) bold italic type letters $\boldsymbol{a}, \ldots, \boldsymbol{z}\,(\boldsymbol{A}, \ldots, \boldsymbol{Z})$ denote vectors (matrices) with interval components. By $IV_N(\mathbb{R})$ we denote vectors $\boldsymbol{x} = (\mathrm{X}_i)_{i=1}^N$ with N interval components and by $IM_{M,N}(\mathbb{R})$ $M \times N$ interval matrices $\boldsymbol{X} = (\mathrm{X}_{i,j})_{i=1,j=1}^{M,N}$. Small (capital) non-bold italic type letters $a, \ldots, z\,(A, \ldots, Z)$ denote *point* vectors (matrices) with real numbers as components.

Addition, subtraction, multiplication, division of intervals are defined as operations on the bounds [1]: $\mathrm{X} + \mathrm{Y} := \{\mathrm{x} + \mathrm{y} \,|\, \mathrm{x} \in \mathrm{X}, \mathrm{y} \in \mathrm{Y}\} = [\underline{\mathrm{X}} + \underline{\mathrm{Y}}, \overline{\mathrm{X}} + \overline{\mathrm{Y}}]$, $\mathrm{X} * \mathrm{Y} := \{\mathrm{x} * \mathrm{y} \,|\, \mathrm{x} \in \mathrm{X}, \mathrm{y} \in \mathrm{Y}\} = [\min\{\underline{\mathrm{X}} * \underline{\mathrm{Y}}, \underline{\mathrm{X}} * \overline{\mathrm{Y}}, \overline{\mathrm{X}} * \underline{\mathrm{Y}}, \overline{\mathrm{X}} * \overline{\mathrm{Y}}\}, \max\{\underline{\mathrm{X}} * \underline{\mathrm{Y}}, \underline{\mathrm{X}} * \overline{\mathrm{Y}}, \overline{\mathrm{X}} * \underline{\mathrm{Y}}, \overline{\mathrm{X}} * \overline{\mathrm{Y}}\}]$.

The real, i.e. point vectors (similarly matrices) of lower bounds, upper bounds, widths, midpoints, absolute values, distances are defined componentwise: $\underline{\boldsymbol{x}} = (\underline{\mathrm{X}_i})_{i=1}^N$, $\overline{\boldsymbol{x}} = (\overline{\mathrm{X}_i})_{i=1}^N$, $d(\boldsymbol{x}) = (d(\mathrm{X}_i))_{i=1}^N$, $m(\boldsymbol{x}) = (m(\mathrm{X}_i))_{i=1}^N$, $|\boldsymbol{x}| = (|\mathrm{X}_i|)_{i=1}^N$, $q(\boldsymbol{x}, \boldsymbol{y}) = (q(\mathrm{X}_i, \mathrm{Y}_i))_{i=1}^N \in \mathbb{R}^N$ for $\boldsymbol{x} = (\mathrm{X}_i)_{i=1}^N, \boldsymbol{y} = (\mathrm{Y}_i)_{i=1}^N \in IV_N(\mathbb{R})$, e.g. Interval operations are continuous as they are defined by continous (point) operations on their bounds.

Enclosures $\boldsymbol{x} \subset \boldsymbol{y}$ and intersections $\boldsymbol{x} \cap \boldsymbol{y}$ of interval vectors are defined componentwise: $\boldsymbol{x} \subset \boldsymbol{y} \equiv \mathrm{X}_i \subset \mathrm{Y}_i \ \forall i \in \{1, \ldots, N\}$ and $\boldsymbol{x} \cap \boldsymbol{y} = (\mathrm{X}_i \cap \mathrm{Y}_i)_{i=1}^N$. For point vectors $x, y \in \mathbb{R}^N$, we define the partial ordering $x < y \equiv \mathrm{x}_i < \mathrm{y}_i \ \forall i \in \{1, \ldots, N\}$. Similarly definitions apply for matrices and other operators. For systems of equations $(\boldsymbol{A}, \boldsymbol{y})$ with interval coefficients $\boldsymbol{A} \in IM_{N,N}(\mathbb{R})$, $\boldsymbol{y} \in IV_N(\mathbb{R})$ and linear form we compute enclosures $\boldsymbol{x} := LES(\boldsymbol{A}, \boldsymbol{y}) \in IV_N(\mathbb{R})$ of the set of solutions $SOL(\boldsymbol{A}, \boldsymbol{y}) := \{x \in \mathbb{R}^N \,|\, \exists\, A \in \boldsymbol{A}, y \in \boldsymbol{y} : Ax = y\}$ of all systems of linear equations $Ax = y$ which can be defined by choosing componentwise an arbitrary point matrix $A \in \boldsymbol{A}$ and an arbitrary point vector $y \in \boldsymbol{y}$ and where LES denotes a suitable interval method. In most cases, $SOL(\boldsymbol{A}, \boldsymbol{y}) \subset \mathbb{R}^N$ is not an interval vector [1].

3 Basics

We consider the basic parabolic problem

$$\begin{array}{lll} u_t - Lu + \phi(u) = 0 & & \Omega \times [0,T], \Omega \subset \mathbb{R}^m, m = 2,3 \\ u(.,t) & = g(.,t) & t \in [0,T] \\ u(.,0) & = u_0 & \text{on } \Omega. \end{array} \tag{1}$$

The discretization with difference methods yields nonlinear systems of the form

$$f(x) \equiv Ax + \psi(x) = 0,\ A \in \mathbb{R}^{N \times N},\ \psi \text{ diagonal: } \psi(x) = (\psi_i(\mathrm{x}_i))_{i=1}^N, \tag{2}$$

which have to be solved in each of a finite number of time steps t and whose solution is unique under appropriate conditions [4].

We derive Newton-like interval methods based on the following strategy: Let be given a nonlinear system of equations $f(x^*) = 0$ in $\mathbb{R}^N$ with a unique solution x^* and where f is Fréchet differentiable. We assume that an initial enclosure $\boldsymbol{x}^0$ of x^* can be computed: $\boldsymbol{x}^0 \ni x^*$ with $f(x^*) = 0$. We compute a sequence of interval

vectors $\boldsymbol{x}^k, k = 0, 1, 2, \ldots$, with $\boldsymbol{x}^k \supseteq \boldsymbol{x}^{k+1} \ni x^* \; \forall\, k \in \mathbb{N}_0$, i.e. which is monotone with respect to componentwise enclosure and whose elements all enclose the solution. The sequence converges monotonically to the solution under appropriate conditions: $\boldsymbol{x}^k \to x^* (k \to \infty)$. Based on splittings $F'(.) = M - (M - F'(.))$ of the Jacobian of f with invertible matrices $M \in \mathbb{R}^{N \times N}$, i.e. with real numbers as entries, we introduce variants of the so-called *Krawczyk-like methods* [3], [7]

$$\boldsymbol{x}^{k+1} := \{m^k - LES(M, (M - \boldsymbol{F'}(\boldsymbol{x}^k))(m^k - \boldsymbol{x}^k) + f^k)\} \cap \boldsymbol{x}^k, k \in \mathbb{N}_0, \quad (3)$$

where $\boldsymbol{F'}(\boldsymbol{x}^k)$ is the interval arithmetic evaluation of $F'(.)$ for an interval vector $\boldsymbol{x}^k$, where $m^k \in \boldsymbol{x}^k$ is a point vector (the vector of midpoints $m(\boldsymbol{x}^k)$, e.g.) and $f^k := f(m^k)$.

The decomposition of the domain Ω and the corresponding index set I for the grid points of the discretized system into overlapping subdomains Ω_j and index subsets I_j, resp., yields the base for parallelism in this context: $\Omega = \bigcup_{j=1}^{p} \Omega_j$ and $I = \{1, \ldots, N\} = \bigcup_{j=1}^{p} I_j$. By definition an overlap and the corresponding grid points in the discretized problem belong to more than one subdomain. In the discretized systems we introduce new variables for the points on overlaps with respect to each subdomain involved in that overlap. We, therefore, define multiple splittings, one for each subdomain. In the sequel, we will derive two-stage iterative methods, thus we define *multiple two-stage splittings*:

$$F'(.) = M_j - N_j(.),\; M_j = P_j - Q_j,\; N_j(.) = M_j - F'(.),\; j = 1, \ldots, p. \quad (4)$$

Based on (3) and (4) we note the following general two-stage Krawczyk-Schwarz method for the solution of a system $f(x^*) = 0$, where we choose $m^{k,j} \in \boldsymbol{x}^{k,j}$ and invertible matrices $M_{k,j}$, $P_{k,j}$ and matrices $Q_{k,j}$ all of which can depend also on the iteration index k. Data exchange and convergence test take place only once in every outer iteration. For each process $p_j, 1 \le j \le p$, an individual sequence of iterates $\boldsymbol{x}^{k,j}, k \in \mathbb{N}_0$, is computed which is expected to converge to the solution x^* of the global problem. As, in practice, we only get enclosures, we can take the intersection $\boldsymbol{x}^*$ of all final iterates $\boldsymbol{x}^{last,j}$ as resulting enclosure.

$$\begin{array}{l}
\textbf{for } j := 1 \textbf{ to } p \textbf{ do parallel} \qquad (5)\\
\quad \boldsymbol{x}^{0,j} := \boldsymbol{x}^0\\
\quad\quad \textbf{for } k := 0 \textbf{ step } 1 \textbf{ until convergence do}\\
\quad\quad\quad \text{data exchange and convergence test}\\
\quad\quad\quad \boldsymbol{x}^{k,0,j} := \boldsymbol{x}^{k,j}\\
\quad\quad\quad \textbf{for } i := 0 \textbf{ to } l - 1 \textbf{ do}\\
\quad\quad\quad\quad \boldsymbol{x}^{k,i+1,j} := \{m^{k,j} - LES(P_{k,j}, (M_{k,j} - \boldsymbol{F'}(\boldsymbol{x}^{k,j}))(m^{k,j} - \boldsymbol{x}^{k,j})\\
\quad\quad\quad\quad\quad\quad + Q_{k,j}(m^{k,j} - \boldsymbol{x}^{k,i,j}) + f(m^{k,j})\} \cap \boldsymbol{x}^{k,i,j}\\
\quad\quad\quad \boldsymbol{x}^{k+1,j} := \boldsymbol{x}^{k,l,j}\\
\boldsymbol{x}^* := \bigcap_{j=1}^{p} \boldsymbol{x}^{last,j}.
\end{array}$$

4 Method

We now return to (1),(2). A fully implicit discretization of (1) yields in each time step $t \in \{1, ..., T/\Delta t\}$ a system (2) of the form

$$\begin{aligned} A^{(t)}x + \psi^{(t)}(x) = 0,\ A^{(t)} &= \tfrac{1}{r}I + L^{(t)}, \\ \psi^{(t)}(x) \quad &= h^2\phi^{(t)}(x) - \tfrac{1}{r}u^{t-1} - b^t \end{aligned} \tag{6}$$

with the solution u^t and where Δt: time step, h: space step and $r := \frac{\Delta t}{h^2}$. The vector b^t contains all terms depending on the initial values and the boundary values of the t-th time step, i.e. all terms not depending on the unknowns in that step, while u^{t-1} denotes the exact solution of the preceding time step $t-1$. Instead of (4), we use two-stage splittings $\forall\, t \in \{1, ..., T/\Delta t\}\ \forall\, j \in \{1, ..., p\}$

$$\begin{aligned} &L^{(t)} = M_j^{(t)} - N_j^{(t)},\ M_j^{(t)}(.) = M_j^{(t)} + h^2(\phi_j^{(t)})'(.), \\ &M_j^{(t)} = P_j^{(t)} - Q_j^{(t)},\ P_j^{(t)}(.) = P_j^{(t)} + h^2(\phi_j^{(t)})'(.), \\ &(\phi_j^{(t)})'(.) = \Phi_j^{(t)} - \{\Phi_j^{(t)} - (\phi_j^{(t)})'(.)\} \end{aligned} \tag{7}$$

with diagonal matrices $(\phi_j^{(t)})'(.), \Phi_j^{(t)}$. We now define our main algorithm in which a two-stage Krawczyk-Schwarz Method (5) adapted to (6) is carried out in every time step t. For every t, the enclosure $\boldsymbol{u}^t$ of the preceding time step appears on the right-hand side of the systems of linear form. This aspect strongly influences the properties of the method. In the following algorithm, we assume

$$\Phi_j^{(t)} \equiv \Phi_{j,k}^{(t)} \in (\boldsymbol{\phi}_j^{(t)})'(\boldsymbol{x}^{k,j}), m^{k,j} \in \boldsymbol{x}^{k,j}.$$

A possible simple choice is given by midpoints:

$$\Phi_j^{(t)} \equiv \Phi_{j,k}^{(t)} := m((\phi_j^{(t)})'(\boldsymbol{x}^{k,j})), m^{k,j} := m(\boldsymbol{x}^{k,j}).$$

for $t := 0$ **until** $T/\Delta t - 1$ **do** (8)

compute an initial enclosure $\boldsymbol{x}^0$ of u^{t+1} from $\boldsymbol{u}^t$

for $j := 1$ **to** p **do parallel**

$\boldsymbol{x}^{0,j} := \boldsymbol{x}^0$

for $k := 0$ **step** 1 **until convergence do**

exchange of artificial boundaries and convergence test

$\boldsymbol{x}^{k,0,j} := \boldsymbol{x}^{k,j}$

$$
\begin{aligned}
&\textbf{for } i := 0 \textbf{ to } l-1 \textbf{ do}\\
&\boldsymbol{x}^{k,i+1,j} := \{m^{k,j} - LES(\tfrac{1}{r}I + P_j^{(t+1)} + h^2\Phi_{j,k}^{(t+1)},\\
&\qquad h^2(\Phi_{j,k}^{(t+1)} - (\phi_j^{(t+1)})'(\boldsymbol{x}^{k,j}))(m^{k,j} - \boldsymbol{x}^{k,j})\\
&\qquad +(\tfrac{1}{r}I + P_j^{(t+1)})m^{k,j} - N_j^{(t+1)}\boldsymbol{x}^{k,j} - Q_j^{(t+1)}\boldsymbol{x}^{k,i,j}\\
&\qquad +h^2\phi_j^{(t+1)}(m^{k,j}) - \tfrac{1}{r}\boldsymbol{u}^t - b^{t+1,j}\} \cap \boldsymbol{x}^{k,i,j}\\
&\boldsymbol{x}^{k+1,j} := \boldsymbol{x}^{k,l,j}
\end{aligned}
$$

$$
\boldsymbol{u}^{t+1} := \bigcap_{j=1}^{p} \boldsymbol{x}^{last,j}.
$$

In each process $p_j, 1 \le j \le p$, of (8) one still formally treats a problem whose size is equal to the original problem. This is inefficient in practice. As our intention is to develop a method, in which each process corresponds to one subdomain including its overlaps with other subdomains, we further modify the splittings (7) used in (8): We redefine

$$
P_j^{(t)}(.) := \begin{pmatrix}
\ddots & & & & \\
& (M_j^{(t)})_{l,l}(.) & & 0 & \\
& & \ddots & & \\
& & & \hat{P}_j^{(t)}(.) & \\
& & & & \ddots \\
& & 0 & & (M_j^{(t)})_{l,l}(.) \\
& & & & & \ddots
\end{pmatrix}, j = 1, \ldots, p, l \notin I_j, \tag{9}
$$

as an $N \times N$ block diagonal matrix. The $|I_j| \times |I_j|$ diagonal block $\hat{P}_j^{(t)}(.), j = 1, \ldots, p$, corresponds to the index set of the j-th domain. We extend $\hat{P}_j^{(t)}(.)$ to $P_j^{(t)}(.)$ by filling its diagonal with the diagonal elements $(M_j^{(t)})_{l,l}(.), l \notin I_j$, of $M_j^{(t)}(.)$. We complete the partial splitting $((M_j^{(t)})_{i,l}(.))_{i\in I_j, l=1,,N} = \hat{P}_j^{(t)}(.) - \hat{Q}_j^{(t)}(.)$ by an appropriate rectangular $|I_j| \times N$ matrix $\hat{Q}_j^{(t)}(.)$ which is extended to an $N \times N$ matrix $Q_j^{(t)}(.)$. Similarly we define $M_j^{(t)}(.), \hat{M}_j^{(t)}(.), N_j^{(t)}(.), \hat{N}_j^{(t)}(.), (\phi_j^{(t)})'(.), (\hat{\phi}_j^{(t)})'(.), \Phi_{j,k}^{(t)}, \hat{\Phi}_{j,k}^{(t)}$. We now rewrite the new iterates $\boldsymbol{x}^{k+1,j}, k = 0, 1, \ldots$ in each step of (8): we replace the next to the last line in (8) by $\hat{\boldsymbol{x}}^{k+1,j} := \boldsymbol{x}^{k,l,j}$ and compute

$$
\boldsymbol{x}^{k+1,j} := \sum_{m=1}^{p} E_{j,m}\hat{\boldsymbol{x}}^{k+1,m}, \quad j = 1, \ldots, p, \tag{10}
$$

with diagonal weighting matrices $E_{j,m}, 1 \le j, m \le p$, defined by $\sum_{m=1}^{p} E_{j,m} = I$, $(E_{j,j})_{l,l} = 0$, if $l \notin I_j$, $(E_{j,m})_{l,l} \in \{0, 1\}$ for $j \ne m$; $E_{j,j}$ "projection" on I_j.

The weighting matrices are defined such that each processor p_j only modifies the components corresponding to its index set I_j, including all grid points from overlaps with neighbouring domains. According to the definition of the weighting matrices, the sum in (10) is nothing but a selection. The definition of the weighting matrices can be generalized in order to get multisplitting-like or more general weighted Schwarz-type methods in the sense of [2].

We have derived a Newton-like interval method of the Krawczyk-type which is based on the Schwarz alternative procedure (SAP) [5], [6]. The data exchange in (8) takes place again only once in every outer step (k-loop) and is restricted to the artificial boundary values of overlaps and the data for a convergence test. In (8), (10) each processor p_j involved in an overlap keeps its own iterates $\boldsymbol{x}^{k,j}$ on that overlap until the end of the iteration process after which the intersection of the final iterates can be carried out to get a final enclosure.

5 Convergence and Enclosure

For the proof of convergence and enclosure properties, we rewrite (8) as a formally multiple iteration in $\mathbb{R}^{pN}$:

$$\textbf{for } t := 1 \textbf{ step } 1 \textbf{ until } T/\Delta t - 1 \textbf{ do} \tag{11}$$

$$\mathbf{x}^0 \equiv \mathbf{x}^0(\mathbf{u}^t)$$

$$\textbf{for } k := 0 \textbf{ step } 1 \textbf{ until convergence do}$$

$$\mathbf{x}^{k+1} := \mathrm{h}^{t+1,k+1}(\mathbf{x}^k)$$

$$\mathbf{u}^{t+1} \equiv (\boldsymbol{u}^{t+1}, \ldots, \boldsymbol{u}^{t+1});\ \boldsymbol{u}^{t+1} := \bigcap_{j=1}^{p} \boldsymbol{x}^{last,j}.$$

We write $\mathbf{u}^t \equiv (\boldsymbol{u}^t, \ldots, \boldsymbol{u}^t), \mathrm{u}^t \equiv (u^t, \ldots, u^t), \mathbf{x}^k = (\boldsymbol{x}^{k,1}, \ldots, \boldsymbol{x}^{k,p}) \in IV_{pN}(\mathbb{R}), \mathrm{h}^{t,k} = (h^{t,k,1}, \ldots, h^{t,k,p}) : IV_{pN}(\mathbb{R}) \to IV_{pN}(\mathbb{R})$.

Theorem 1. *The following assertions hold for* $\forall\, t \in \{1, \ldots, T/\Delta t\}$ *and* $\forall\, j \in \{1, \ldots, p\}$. *If (1)* $u^t \in \boldsymbol{x}^0 \equiv \boldsymbol{x}^{0,t}$, *(2)* ϕ *is Fréchet differentiable on* $\boldsymbol{x}^0$, *(3)* $L^{(t)}, M_j^{(t)}, P_j^{(t)}$ *are M matrices, (4)* $(\phi^{(t)})'(\boldsymbol{x}^0), N_j^{(t)}, Q_j^{(t)} \geq 0$, *(5) LES is applicable to* $P_j(\boldsymbol{x}^0) := \frac{1}{r} I + P_j^{(t)} + h^2(\phi_j^{(t)})'(\boldsymbol{x}^0)$, *(6)*$\forall P \in P_j(\boldsymbol{x}^0)\, \forall\, \boldsymbol{x} \in IV_N(\mathbb{R}) :$ $LES(P, \boldsymbol{x}) = P^{-1}\boldsymbol{x}$, *(7)* $\Phi_{j,k}^{(t)} \equiv \Phi_j^{(t)}(\boldsymbol{x}^{k,j}) := m((\phi_j^{(t)})'(\boldsymbol{x}^{k,j})), m^{k,j} \in \boldsymbol{x}^{k,j}$,

then (1) (monotone enclosure) $\forall\, k \in \mathbf{N}_0 : u^t \in \boldsymbol{u}^t \subseteq \boldsymbol{x}^{k+1,j} \subseteq \boldsymbol{x}^{k,j}$;

If $\boldsymbol{u}^t \equiv u^t$ *then (2a) (convergence to the unique solution)* $\lim_{k\to\infty} \boldsymbol{x}^{k,j} = u^{t+1}$, *(2b) (asymptotic convergence speed)* $R_1(\mathrm{u}^{t+1}) \leq \rho(\mathrm{H}_l^{(t)}(\mathrm{u}^{t+1})) < 1$, *where*

$$\mathrm{H}_l^{(t)}(\mathrm{u}^t) = \begin{pmatrix} E_{1,1}T_1^{(t)}(u^t) \cdots E_{1,p}T_p^{(t)}(u^t) \\ \vdots \qquad\qquad \vdots \\ E_{p,1}T_1^{(t)}(u^t) \cdots E_{p,p}T_p^{(t)}(u^t) \end{pmatrix},$$

$$T_j^{(t)}(u^t) = ((\tilde{P}_j^{(t)})^{-1}Q_j^{(t)})^l + \textstyle\sum_{i=0}^{l-1}((\tilde{P}_j^{(t)})^{-1}Q_j^{(t)})^i(\tilde{P}_j^{(t)})^{-1}N_j^{(t)},$$

$$\tilde{P}_j^{(t)} \quad = \tfrac{1}{r}I + P_j^{(t)} + h^2(\phi_j^t)'(u^t),$$

(2c) (number of inner iterations) $\rho(\mathrm{H}_m^{(t)}(\mathrm{u}^t)) \leq \rho(\mathrm{H}_l(\mathrm{u}^t))$ *for* $l \leq m$.

If $d(\boldsymbol{u}^t) \neq 0$ *then (3a) (convergence to an enclosure)* $\exists\, \boldsymbol{x}_j^* \in IV_N(\mathbb{R})$: $\lim_{k\to\infty} \boldsymbol{x}^{k,j} = \boldsymbol{x}_j^*$ *and* $\boldsymbol{u}^{t+1} := \bigcap_{j=1}^{p} \boldsymbol{x}_j^*$*; (3b) (width) Define*

$$\rho_t := \min_{j=1}^{p} \rho((I + rL^{(t+1)} + \Delta t\, \underline{(\phi_j^{t+1})'(\boldsymbol{x}_j^*)})^{-1}).$$

If $L^{(t+1)}$ *is also positive definite* <u>*then*</u> $\forall\, \epsilon \in (0, 1-\rho_t)\, \exists$ *monotone* $\|.\|_\epsilon$ *:*

$$\|d(\boldsymbol{u}^{t+1})\|_\epsilon \leq \rho_t \|d(\boldsymbol{u}^t)\|_\epsilon < \|d(\boldsymbol{u}^t)\|_\epsilon$$

If, in addition, $L^{(t+1)}$ *is symmetric then*

$$\|d(\boldsymbol{u}^{t+1})\|_2 \leq \rho_t \|d(\boldsymbol{u}^t)\|_2 < \|d(\boldsymbol{u}^t)\|_2.$$

Proof. (1) The enclosures $\boldsymbol{u}^t \subseteq \boldsymbol{x}^{k+1,j} \subseteq \boldsymbol{x}^{k,j}$ hold by definition. The proof proceeds by induction on k and i and fixed t. In the sequel, we omit the index $t+1$ for all variables but u for better readability. Assume now $u^t \in \boldsymbol{u}^t, u^{t+1} \in \boldsymbol{x}^{k,i,j}$ for some k, i and all $j \in \{1, \ldots, p\}$. By assumption this is true for $i, k = 0$. Let be C an arbitrary nonsingular point matrix. There exists $z \in \boldsymbol{x}^{k,j}$ such that

$$\begin{aligned} f(m^{k,j}) = f(m^{k,j}) - f(u^{t+1}) &= f'(z)(m^{k,j} - u^{t+1}) \\ &= (f'(z) - C)(m^{k,j} - u^{t+1}) + C(m^{k,j} - u^{t+1}), \end{aligned}$$

or

$$u^{t+1} = m^{k,j} - C^{-1}\left((C - f'(z))(m^{k,j} - u^{t+1}) + f(m^{k,j})\right).$$

Choosing $C := \frac{1}{r}I + P_j + h^2\Phi_j$ we get

$$\begin{aligned} u^{t+1} = m^{k,j} &- \left(\tfrac{1}{r}I + P_j + h^2\Phi_j\right)^{-1} \\ &\left\{\left(h^2(\Phi_j - \phi_j'(z)) + N_j + Q_j\right)(m^{k,j} - u^{t+1})\right. \\ &\left. + \left(\tfrac{1}{r}I + P_j - Q_j - N_j\right)m^{k,j} + h^2\phi_j(m^{k,j}) - \tfrac{1}{r}u^t - b_j\right\} \\ = m^{k,j} &- \left(\tfrac{1}{r}I + P_j + h^2\Phi_j\right)^{-1} \\ &\left\{\left(h^2(\Phi_j - \phi_j'(z))\right)(m^{k,j} - u^{t+1})\right. \\ &\left. +(\tfrac{1}{r}I + P_j)m^{k,j} - (Q_j + N_j)u^{t+1} + h^2\phi_j(m^{k,j}) - \tfrac{1}{r}u^t - b_j\right\}. \end{aligned}$$

Using the definition of LES and $u^t \in \boldsymbol{u}^t, u^{t+1} \in \boldsymbol{x}^{k,i,j} \subseteq \boldsymbol{x}^{k,0,j} \equiv \boldsymbol{x}^{k,j}$, we obtain

$$\begin{aligned} u^{t+1} \in m^{k,j} - LES\big(&\tfrac{1}{r}I + P_j + h^2\Phi_j\ , \\ &h^2(\Phi_j - \phi_j'(\boldsymbol{x}^{k,j}))(m^{k,j} - \boldsymbol{x}^{k,j}) + (\tfrac{1}{r}I + P_j)m^{k,j} - N_j\boldsymbol{x}^{k,j} \\ &-Q_j\boldsymbol{x}^{k,i,j} + h^2\phi_j(m^{k,j}) - \tfrac{1}{r}\boldsymbol{u}^t - b_j\big) \cap \boldsymbol{x}^{k,i,j} =: \boldsymbol{x}^{k,i+1,j}. \end{aligned}$$

hence (1) is proved. Due to intersection and the continuity of interval operations [1], we can deduce (3a): there exists $\boldsymbol{x}_j^* \in IV_N(\mathbb{R})$ for all $j \in \{1,\ldots,p\}, i \in \{1,\ldots,l\}$

$$u^{t+1} \in \boldsymbol{x}_j^* := \lim_{k\to\infty} \boldsymbol{x}^{k,j} = \lim_{k\to\infty} \boldsymbol{x}^{k,i,j}.$$

We now consider the width $d(.)$. With $d(\boldsymbol{AB}) = |\boldsymbol{A}|d(\boldsymbol{B})$ for $N \times N$ interval matrices $\boldsymbol{A}, \boldsymbol{B}$, if $0 \in \boldsymbol{B}$ or $\boldsymbol{A} \equiv A$, and assumption (6) we get

$$\begin{aligned} d(\boldsymbol{x}^{k,i+1,j}) \le &\big(\tfrac{1}{r}I + P_j + h^2\Phi_j\big)^{-1} \\ &\big(\big(h^2|\Phi_j - \phi_j'(\boldsymbol{x}^{k,j})| + N_j\big)\, d(\boldsymbol{x}^{k,j}) + Q_j d(\boldsymbol{x}^{k,i,j}) + \tfrac{1}{r}d(\boldsymbol{u}^t)\big)\,. \end{aligned} \tag{12}$$

For $u^t \equiv \boldsymbol{u}^t$, i.e. $d(\boldsymbol{u}^t) = d(u^t) = 0$, and remembering $\Phi_j = m(\phi_j'(\boldsymbol{x}^{k,j}))$ we deduce

$$\begin{aligned} d(\boldsymbol{x}^{k,i+1,j}) &\le \big(\tfrac{1}{r}I + P_j + h^2 m(\phi_j'(\boldsymbol{x}^{k,j}))\big)^{-1} \\ &\quad \big(\big(h^2|m(\phi_j'(\boldsymbol{x}^{k,j})) - \phi_j'(\boldsymbol{x}^{k,j})| + N_j\big)\, d(\boldsymbol{x}^{k,j}) + Q_j d(\boldsymbol{x}^{k,i,j})\big) \\ &= \big(\tfrac{1}{r}I + P_j + h^2 m(\phi_j'(\boldsymbol{x}^{k,j}))\big)^{-1} \\ &\quad ((h^2(m(\phi_j'(\boldsymbol{x}^{k,j})) - \underline{\phi_j'(\boldsymbol{x}^{k,j})}) + N_j) d(\boldsymbol{x}^{k,j}) + Q_j d(\boldsymbol{x}^{k,i,j})). \end{aligned} \tag{13}$$

Due to the continuity of interval operations and the involved functions, this inequality also holds for the limits $\boldsymbol{x}_j^*$:

$$\begin{aligned} d(\boldsymbol{x}_j^*) \le &\big(\tfrac{1}{r}I + P_j + h^2 m(\phi_j'(\boldsymbol{x}_j^*))\big)^{-1} \\ &\Big(h^2(m(\phi_j'(\boldsymbol{x}_j^*)) - \underline{\phi_j'(\boldsymbol{x}_j^*)}) + N_j\ + Q_j\Big)\, d(\boldsymbol{x}_j^*). \end{aligned} \tag{14}$$

We first note

$$\begin{aligned} &\tfrac{1}{r}I + P_j + h^2 m(\phi_j'(\boldsymbol{x}_j^*)) - ((h^2(m(\phi_j'(\boldsymbol{x}_j^*)) - \underline{\phi_j'(\boldsymbol{x}_j^*)}) + N_j + Q_j) \\ &= \tfrac{1}{r}I + L + h^2\underline{\phi_j'(\boldsymbol{x}_j^*)}. \end{aligned} \tag{15}$$

L, P_j are M matrices , $m(\phi_j'(\boldsymbol{x}_j^*) \ge \underline{\phi_j'(\boldsymbol{x}_j^*)} \ge 0$. Therefore, the matrices $B := \frac{1}{r}I + L + h^2\underline{\phi_j'(\boldsymbol{x}_j^*)}$, $C := \frac{1}{r}I + P_j + h^2 m(\phi_j'(\boldsymbol{x}_j^*))$ are also M matrices, i.e. they have a componentwise nonnegative inverse

$$\left(\frac{1}{r}I + P_j + h^2 m(\phi_j'(\boldsymbol{x}_j^*))\right)^{-1}, \left(\frac{1}{r}I + L + h^2\underline{\phi_j'(\boldsymbol{x}_j^*)}\right)^{-1} \ge 0 \tag{16}$$

We further note

$$D := h^2(m(\phi_j'(\boldsymbol{x}_j^*)) - \underline{\phi_j'(\boldsymbol{x}_j^*)}) + N_j + Q_j \geq 0. \tag{17}$$

Now $B = C - D$, B, C are M matrices and $D \geq 0$, hence $B = C - D$ is a regular splitting, i.e. $E := C^{-1}D \geq 0$ and $\rho(E) < 1$ [9]. For a nonnegative matrix with spectral radius less than one we know that $I - E$ has a nonnegative inverse [9]. From (12) we now get

$$0 \leq (I - E)^{-1} d(\boldsymbol{x}_j^*) \leq 0, \tag{18}$$

i.e. $d(\boldsymbol{x}_j^*) = 0$ or $\boldsymbol{x}_j^* \equiv x_j^*$. With continuity arguments $u^{t+1} \in \boldsymbol{x}^{k,j}$ for all k, j also yields $u^{t+1} \in \boldsymbol{x}_j^* \equiv x_j^*$, hence $\boldsymbol{u}^{t+1} \equiv u^{t+1} = x_j^*$ for all $j \in \{1, \ldots, p\}$. This proves (2a). The assertions (2b) and (2c) now follow from (13) by standard arguments for two stage and multistep iterative methods.

If $d(\boldsymbol{u}^t) \neq 0$, we deduce from (12) similarly to (14)

$$\begin{aligned} d(\boldsymbol{x}_j^*) \leq & \left(\tfrac{1}{r} I + P_j + h^2 m(\phi_j'(\boldsymbol{x}_j^*))\right)^{-1} \\ & \left\{ \left(h^2(m(\phi_j'(\boldsymbol{x}_j^*)) - \underline{\phi_j'(\boldsymbol{x}_j^*)}) + N_j \ + Q_j \right) d(\boldsymbol{x}_j^*) + \tfrac{1}{r} d(\boldsymbol{u}^t) \right\}. \end{aligned} \tag{19}$$

Repeating the arguments leading from (14) to (18) we get

$$(I - E)^{-1} d(\boldsymbol{x}_j^*) \leq C^{-1} \frac{1}{r} d(\boldsymbol{u}^t), \tag{20}$$

hence

$$d(\boldsymbol{x}_j^*) \leq (I - C^{-1}D)^{-1} C^{-1} \frac{1}{r} d(\boldsymbol{u}^t) = (C - D)^{-1} \frac{1}{r} d(\boldsymbol{u}^t) = B^{-1} \frac{1}{r} d(\boldsymbol{u}^t), \tag{21}$$

or

$$d(\boldsymbol{u}^{t+1}) \leq d(\boldsymbol{x}_j^*) \leq \left(I + rP_j + \Delta t \, \underline{\phi_j'(\boldsymbol{x}_j^*)} \right)^{-1} d(\boldsymbol{u}^t). \tag{22}$$

The first norm inequality of assertion (3b) now follows immediately from [1], Ch. 12, Th. 1, the second inequlaity for symmetric $L^{(t+1)}$ is obvious.

We have discussed only methods for systems (6) resulting from a fully implicit discretization of (1), e.g. Theoretically, other discretizations like Crank-Nicolson could be used. In this case the term $\frac{1}{r}\boldsymbol{u}^t$ in (8) is replaced by $(\frac{1}{r} I + L^{(t+1)})\boldsymbol{u}^t$. This term frequently leads to overestimating enclosures. The convergence to the (point) solution can be shown only for inefficient values of r. The relation $LES(A, \boldsymbol{x}) = A^{-1}\boldsymbol{x}$ for point matrices A and interval vectors $\boldsymbol{x}$ required in condition (6) exists for all standard methods LES (see [8] for examples). The assumption $\boldsymbol{u}^t \equiv u^t$ leading to the assertions (2a), (2b), (2c) means, that in every time step one theoretically gets the exact solution $\boldsymbol{u}^{t+1} \equiv u^{t+1}$ of the respective time step under the condition that the exact solution of the previous time step is known. We then also obtain an estimate of the asymptotic convergence factor. In practice, one only gets, due to rounding errors, an enclosure of the exact solution: $\boldsymbol{u}^{t+1} \ni u^{t+1}$, i.e. $d(\boldsymbol{u}^{t+1}) \geq 0$. This positive width propagates from one

time step to another. We end up with a final enclosure $\boldsymbol{x}^*$ with positive width, but at least for positive definite or for symmetric matrices $L^{(t)}$, we can prove that the sequence of widths strictly decreases in time by a factor $\rho < 1$ in adequate norms. Condition (7) can be modified for other definitions of $\Phi_j^{(t)}$ and $m^{k,j}$, which require slightly modified proofs based on elementary rules for interval operations. As an example we mention $\Phi_j^{(t)} \in (\phi_j^{(t)})'(\boldsymbol{x}^{k,j}), m^{k,j} := m(\boldsymbol{x}^{k,j})$.

As mentioned already in the second section, we need an initial enclosure $\boldsymbol{x}^0$ of the solution. For systems (2) $Ax + \psi(x) = 0$, such an interval vector can be computed from $\boldsymbol{x}^0 \supseteq A^{-1}\,|\psi(0)|\,[-1,1] \ni x^*$ [1], [4]. In the present context, we get

$$\boldsymbol{x}^0 \supseteq LES\left(\frac{1}{r}I + L^{(t+1)}, \left|h^2\phi^{(t+1)}(0) - \frac{1}{r}\boldsymbol{u}^t + b^{t+1}\right|[-1,1]\right), \tag{23}$$

or, for symmetric $L^{(t+1)}$ even simpler,

$$\boldsymbol{x}^0 \supseteq \rho\left((\frac{1}{r}I + L^{(t+1)})^{-1}\right)\left\|\left|h^2\phi^{(t+1)}(0) - \frac{1}{r}\boldsymbol{u}^t + b^{t+1}\right|\right\|_2 [-1,1]. \tag{24}$$

6 Numerical Example

The numerical tests have been carried out on a Linux cluster with 32 Sun V20z (each with two processors AMD Opteron 252 (2.6 GHz) and 4 GB of main memory) and an Infiniband network. The interval arithmetic is realized by simulation of an outsided rounding in FORTRAN (see [1] for the principle). We consider the example

$$\begin{aligned}
&u_t + ((1.33 + t/10)u_x)_x + ((1.21 + t/5)u_y)_y + e^{tu/10} + u = 0 \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{on } \Omega\times\Theta = [0,1]\times[0,s]\times[0,1] \\
&u(x,y,t) = \begin{cases} \frac{2y}{q+1} + t & \text{for } x = 0 \\ \frac{p+1+2y}{q+1} + 2.1t & \text{for } x = 1 \\ \frac{x}{q+1} + 2 + 4.1t & \text{for } y = s \\ \frac{x}{q+1} + 1.1t & \text{for } y = 0 \end{cases} \quad \text{on } \Gamma_\Omega \\
&u(x,y,0) = 20\sin\left(\{20(x + (y-h)\}n\right) \quad \text{on } \Omega.
\end{aligned} \tag{25}$$

We subdivide the rectangle Ω horizontally into p simply overlapping strips, i.e. each horizontal strip only overlaps with its upper and lower neighbour. We apply a standard five point discretization with central differences in space and fully implicit discretization in time with a mesh size of $\Delta t = h = 1/(n+1)$ in space and time, resp. Let be $s = (q+1)h$, then we get a problem size of $N \times N, N = n * q$ with p subdomains of size $(q/p + 2ov) \times N$. For simplicity, we assume an overlap of ov grid lines between adjacent subdomains. The coefficient

matrices are block tridiagonal with $q \times q$ blocks of size $n \times n$. We apply (8) where we choose Block-Jacobi as inner iteration (the outer iteration is by definition a Block-Jacobi method in additive Schwarz-like methods). We compute the relative distance $q_{rel}(\boldsymbol{x}, \boldsymbol{y})$ and the relative width $d_{\mathrm{rel}}(\boldsymbol{x})$ defined here by

$$\begin{aligned} q_{\mathrm{rel}}(\boldsymbol{x}, \boldsymbol{y}) &:= \max_{1\le i\le n, 1\le j\le q} \left\{ \frac{q\left(\mathrm{Y}_{i,j}, \mathrm{X}_{i,j}\right)}{\max\left\{\left|\mathrm{X}_{i,j}\right|, \left|\mathrm{Y}_{i,j}\right|, 10^{-308}\right\}} \right\}, \\ d_{\mathrm{abs}}(\boldsymbol{x}) &:= \max_{1\le i\le n, 1\le j\le q} \left\{ d(\mathrm{X}_{i,j}) \right\}, \\ d_{rel}(\boldsymbol{x}) &:= \max_{1\le i\le n, 1\le j\le q} \left\{ \frac{d(\mathrm{X}_{i,j})}{\max\left\{\left|\mathrm{X}_{i,j}\right|, 10^{-308}\right\}} \right\}. \end{aligned} \tag{26}$$

We choose the *convergence criterion* $q_{\mathrm{rel}}(\boldsymbol{x}^{k+1}, \boldsymbol{x}^k) < \epsilon$ for all $1 \le j \le p$. In our examples, the matrices are symmetric. We can, therefore, profit from (24) to compute initial enclosures. In the sequel $t_p, S_p := t_1/t_p, E_p := p/S_p$ denote the real time in seconds, the speed up and the efficiency, resp., on p processors. T_s indicates the number of time steps, $it_{\mathrm{out}}, it_{\mathrm{in}}$ the number of outer and inner iterations, resp. In the following examples, we choose $q = 768, n = 500, T_s = 100, N = 384.000$.

Table 1. Parallelism

p	it_{out}	t_p	S_p	E_p
1	37 - 43	1318.0	1.00	1.00
2	37 - 43	637.5	2.07	1.03
4	37 - 47	301.7	4.37	1.09
8	37 - 45	171.5	7,69	0.96
16	37 - 44	84.7	15.56	0.97
32	37 - 47	41.9	31.49	0.98
48	38 - 47	26.9	49.00	1.02

Tab.1 illustrates the parallelism of the method for $\epsilon = 10^{-12}, ov = 1, it_{\mathrm{in}} = 4$. The number it_{out} of outer iterations varies from time step to time step due to the influence of the boundary values. It also varies with respect to the number p of subdomains (processors) because the total system size increases due to an increasing number of additional grid lines ($2ov$ per subdomain), thus the original $N \times N$ system increases to a system of size $(N + (p-1) * ov) \times N$. This explains, in particular, efficiencies E_p greater than 1. In view of this fact, the speed ups S_p and the efficiencies are very satisfactory. For all results in Tab.1, we noted a maximal relative width $d_{\mathrm{rel}}(\boldsymbol{u}^{T_s}) = 0.1 * 10^{-8}$ after the last time step.

By increasing the number of inner iterations one possibly saves a considerable amount of arithmetic work in each outer step, in particular, by reducing the

Table 2. Inner iterations

it_{in}	it_{out}	t_p
1	128 - 163	63.4
2	72 - 87	38.7
3	49 - 60	30.7
4	38 - 47	26.9
5	31 - 39	24.6
6	28 - 34	24.2
7	24 - 33	23.3
8	23 - 30	24.5
9	21 - 29	25.8
10	21 - 29	26.5
11	20 - 24	26.5

number of evaluations of functions, which, on the other hand, are not updated. Tab.2 shows for $\epsilon = 10^{-12}, ov = 1$ that a few inner iterations can drastically reduce the computing time. Increasing their number further leads to a slightly reincreasing time.

Increasing the number of overlapping grid lines increases the size of the subproblems, but also reduces the convergence factor. Tab.3 shows for $p = 48, \epsilon = 10^{-12}, it_{\text{in}} = 4$ that a moderate overlap can also reduce the computation time.

Tab.4 illustrates for the last step T_s the dependence of the relative width $d_{\text{rel}}(\boldsymbol{u}^{T_s})$ and the absolute width $d_{\text{abs}}(\boldsymbol{u}^{T_s})$ on the convergence criterion which is based on a relative distance $q_{\text{rel}}(\boldsymbol{u}^{k+1}, \boldsymbol{u}^k)$ for $p = 48, q = 768, ov = 1, it_{in} = 4$. Theoretically, the exact (point) solution u^t could be computed in each time step. Then one could expect a width decreasing to zero. As we stop the iteration in

Table 3. Overlap

ov	it_{out}	t_p
1	56 - 69	34.8
2	38 - 47	28.0
3	38 - 46	31.1
4	37 - 45	32.6
5	37 - 49	36.5
6	37 - 45	40.6

Table 4. Convergence criterion and width

$q_{\mathrm{rel}}(\boldsymbol{u}^{k+1}, \boldsymbol{u}^k) < \epsilon$	$d_{\mathrm{abs}}(\boldsymbol{u}^{T_s})$	$d_{\mathrm{rel}}(\boldsymbol{u}^{T_s})$	it_{out}	t_p
$10^{-\ 2}$	$0.14 * 10^{-10}$	$0.33 * 10^{-2}$	21 - 27	15.6
$10^{-\ 4}$	$0.33 * 10^{-12}$	$0.73 * 10^{-4}$	24 - 30	17.9
$10^{-\ 6}$	$0.13 * 10^{-13}$	$0.48 * 10^{-6}$	28 - 34	21.4
$10^{-\ 8}$	$0.13 * 10^{-13}$	$0.48 * 10^{-6}$	28 - 34	19.9
10^{-10}	$0.13 * 10^{-13}$	$0.38 * 10^{-9}$	35 - 43	25.5
10^{-12}	$0.13 * 10^{-13}$	$0.31 * 10^{-9}$	38 - 47	27.1
10^{-14}	$0.13 * 10^{-13}$	$0.31 * 10^{-9}$	42 - 55	32.6

each time step at a given precision, we only get an enclosure $d(\boldsymbol{u}^t) \neq 0$. This interval vector appears on the right-hand side in the next time step. Thus, one cannot expect the convergence to a point vector. Therefore, the convergence criterion is based on the relative distance of consecutive iterates. Obviously, the relative width decreases with decreasing distance until a minimum is reached which is due to the usually nonzero width of $\boldsymbol{u}^t$.

References

1. Alefeld, G., Herzberger, J.: Introduction to Interval Computations. Academic Press, New York (1983)
2. Frommer, A., Schwandt, H.: A Unified Representation and Theory of Algebraic Additive Schwarz and Multisplitting Methods. SIAM J. Matrix Anal. Appl. 15, 893–912 (1997)
3. Krawczyk, R.: Newton–Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken. Computing 4, 187–201 (1969)
4. Ortega, J., Rheinboldt, W.: Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York (1970)
5. Rodrigue, G.: Inner/outer iterative methods and numerical Schwarz algorithms. Parallel Comp. 2, 205–218 (1985)
6. Rodrigue, G., Shan, K.L., Yu-Hui, L.: Convergence and comparison analysis of some numerical Schwarz methods. Numer. Math. 56, 123–138 (1989)
7. Schwandt, H.: Krawczyk-like Algorithms for the Solution of Systems of Nonlinear Equations. SIAM J. Num. Anal. 22, 792–810 (1985)
8. Schwandt, H.: The Solution of Nonlinear Elliptic Dirichlet Problems on Rectangles by Almost Globally Convergent Interval Methods. SIAM J. Sc. St. Comp. 6, 617–638 (1985)
9. Varga, R.: Matrix Iterative Analysis. Prentice Hall, Englewood Cliffs (1962)

Red-Black EDG*SOR* Iterative Method Using Triangle Element Approximation for 2D Poisson Equations

J. Sulaiman[1], M. Othman[2], and M.K. Hasan[3]

[1] School of Science and Technology, Universiti Malaysia Sabah, Locked Bag 2073, 88999 Kota Kinabalu, Sabah, Malaysia
[2] Faculty of Computer Science and Info. Tech., Universiti Putra Malaysia, 43400 Serdang, Selangor D.E.
[3] Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor D.E.
jumat@ums.edu.my

Abstract. This paper discusses the use of the 4 Point-Explicit Decoupled Group (EDG) iterative method together with a weighted parameter, namely 4 Point-EDG*SOR*. The effectiveness of this method will be investigated to solve two-dimensional Poisson equations by using the half-sweep triangle finite element approximation equation based on the Galerkin scheme. In fact, formulations of the full-sweep and half-sweep triangle finite element approaches are also shown. Then implementation of the 4 Point-EDG*SOR* was performed by combining the Red-Black (RB) ordering strategy. Some numerical experiments are conducted to show that the 4 Point-EDG*SOR*-RB method is superior to the existing 4 Point-EDG method.

Keywords: Explicit Decoupled Group, Red-Black Ordering, Galerkin Scheme, Triangle Element.

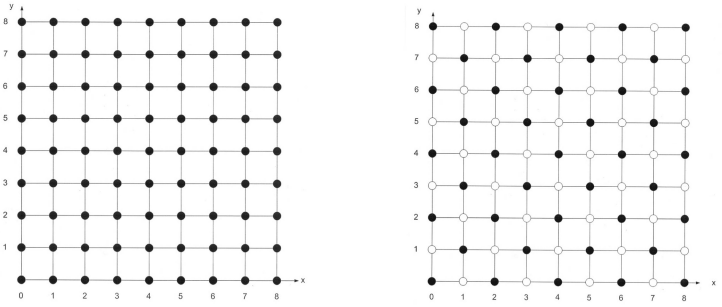
1 Introduction

By using the finite element method, reseachers can consider some weighted residual schemes such as the subdomain, collocation, least-square, moments and Galerkin that can be used to gain approximate solutions (Fletcher [4,5]). By using the first order triangle finite element approximation equation based on the Galerkin scheme, this paper proposes the 4 Point-EDG*SOR* method with the Red-Black (RB) ordering strategy for solving the two-dimensional Poisson equation. In comparison, three iterative methods will be carried out such as the Full-Sweep Gauss-Seidel (FSGS), Half-Sweep Gauss-Seidel (HSGS) and standard EDG.

To investigate the effectiveness of the 4 Point-EDG*SOR*-RB method, let us consider the two-dimensional Poisson equation defined as

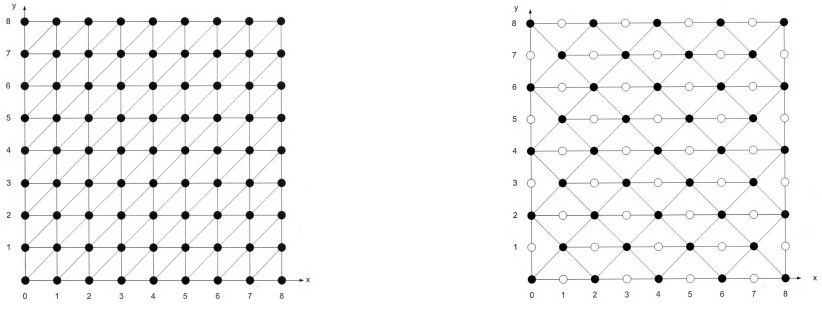
$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = f(x,y), \quad (x,y) \in D = [a,b] \times [a,b] \tag{1}$$

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 298–308, 2007.
© Springer-Verlag Berlin Heidelberg 2007



a) **b)**

Fig. 1. a). and b). show the distribution of uniform node points for the full- and half-sweep cases respectively at $n = 7$



a) **b)**

Fig. 2. a). and b). show the networks of triangle elements for the full- and half-sweep cases respectively at $n = 7$

subject to the Dirichlet boundary conditions

$$\begin{aligned} U(x,a) &= g_1(x), \quad a \le x \le b, \\ U(x,b) &= g_2(x), \quad a \le x \le b, \\ U(a,y) &= g_3(y), \quad a \le y \le b, \\ U(b,y) &= g_4(y), \quad a \le y \le b. \end{aligned}$$

To formulate the full-sweep and half-sweep linear finite element approximation equations for problem (1), we shall restrict our discussion onto uniform node points only as shown in Figure 1. Based on the figure, it has been shown that the solution domain, D need to be discretized uniformly in both x and y directions with a mesh size, h which is defined as

$$h = \frac{b-a}{m}, \quad m = n+1. \tag{2}$$

Based on Figure 1, the networks of triangle finite elements need to be built in order to facilitate us to derive triangle finite element approximation equations for problem (1). Similarly, the concept of the half-sweep iterative applied to the finite difference method (Abdullah [1], Sulaiman *et al.* [13], Othman & Abdullah [8]), each triangle element will involves three node points only of type • as shown in Figure 2. Thus, the implementation of the full-sweep and half-sweep iterative algorithms will be carried out onto the node points of the same type until the iterative convergence test is reached. Then other approximate solutions at remaining points (points of the different type) are computed directly (Abdullah [1], Abdullah & Ali [2], Ibrahim & Abdullah [6], Sulaiman *et al.* [13,14], Yousif & Evans [17]).

2 Half-Sweep Finite Element Approximation

As mentioned in the previous section, this paper will discuss the use of the 4 Point-EDG*SOR*-RB method by using the half-sweep linear finite element approximation equation based on the Galerkin scheme to solve two-dimensional Poisson equations. By considering three node points of type • only, the general approximation of the function, $U(x, y)$ in the form of interpolation function for an arbitrary triangle element, e is given by (Fletcher [4], Lewis & Ward [7], Zienkiewicz [19])

$$\widetilde{U}^{[e]}(x, y) = N_1(x, y)U_1 + N_2(x, y)U_2 + N_3(x, y)U_3 \tag{3}$$

and the shape functions, $N_k(x, y),\ k = 1, 2, 3$ can generally be shown as

$$N_k(x, y) = \frac{1}{\det A}(a_k + b_k x + c_k y), \quad k = 1, 2, 3 \tag{4}$$

where,

$$\det A = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2),$$

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} x_2y_3 - x_3y_2 \\ x_3y_1 - x_1y_3 \\ x_1y_2 - x_2y_1 \end{bmatrix}, \quad \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} y_2 - y_3 \\ y_3 - y_1 \\ y_1 - y_2 \end{bmatrix}, \quad \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} x_3 - x_2 \\ x_1 - x_3 \\ x_2 - x_1 \end{bmatrix}.$$

Beside this, the first order partial derivatives of the shape functions towards x and y are given respectively as

$$\left.\begin{array}{l} \frac{\partial}{\partial x}\left(N_k(x, y)\right) = \frac{b_k}{\det A} \\ \frac{\partial}{\partial y}\left(N_k(x, y)\right) = \frac{c_k}{\det A} \end{array}\right\}, \quad k = 1, 2, 3 \tag{5}$$

In Figure 3, the definition of the hat function,$R_{r,s}(x, y)$ in the solution domain is easily to be shown. Then based on the distribution of the hat function, $R_{r,s}(x, y)$ in the figure, the approximation of the functions, $U(x, y)$ and $f(x, y)$ in case of the full-sweep and half-sweep cases for the entire domain will be defined respectively as (Vichnevetsky [16])

$$\widetilde{U}(x, y) = \sum_{r=0}^{m}\sum_{s=0}^{m} R_{r,s}(x, y)U_{r,s} \tag{6}$$

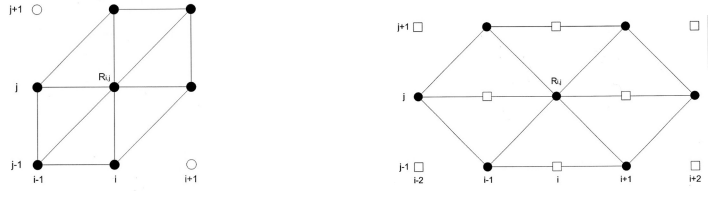
$$\widetilde{f}(x,y) = \sum_{r=0}^{m}\sum_{s=0}^{m} R_{r,s}(x,y) f_{r,s} \tag{7}$$

and

$$\widetilde{U}(x,y) = \sum_{r=0,2,4}^{m}\sum_{s=0,2,4}^{m} R_{r,s}(x,y) U_{r,s} + \sum_{r=1,2,5}^{m-1}\sum_{s=1,3,5}^{m-1} R_{r,s}(x,y) U_{r,s} \tag{8}$$

$$\widetilde{f}(x,y) = \sum_{r=0,2,4}^{m}\sum_{s=0,2,4}^{m} R_{r,s}(x,y) f_{r,s} + \sum_{r=1,3,5}^{m-1}\sum_{s=1,3,5}^{m-1} R_{r,s}(x,y) f_{r,s} \tag{9}$$

Thus, Eqs. (6) and (8) are approximate solutions for problem (1).



a) **b)**

Fig. 3. a). and b). show the definition of the hat function, $R_{i,j}(x,y)$ of triangle elements at the solution domain for the full-sweep and half-sweep cases

To construct the full-sweep and half-sweep linear finite element approximation equations for problem (1), this paper proposes the Galerkin finite element scheme. Thus, let consider the Galerkin residual method (Fletcher [4,5], Lewis & Ward [7]) be defined as

$$\int\int_{D} R_{i,j}(x,y) E(x,y)\, dxdy = 0, \quad i,j = 0,1,2,...,m \tag{10}$$

where, $E(x,y) = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} - f(x,y)$ is a residual function. By applying the Green theorem, Eq. 10 can be shown in the following form

$$\begin{aligned} &\oint_{\lambda} \left(-R_{i,j}(x,y) \frac{\partial U}{\partial y}\, dx + R_{i,j}(x,y) \frac{\partial U}{\partial x}\, dy \right) \\ &- \int_a^b \int_a^b \left(\frac{\partial R_{i,j}(x,y)}{\partial x} \frac{\partial U}{\partial x} + \frac{\partial R_{i,j}(x,y)}{\partial y} \frac{\partial U}{\partial y} \right) dxdy = F_{i,j} \end{aligned} \tag{11}$$

where,

$$F_{i,j} = \int_a^b \int_a^b R_{i,j}(x,y) f(x,y)\, dxdy$$

By applying Eq. (5) and substituting the boundary conditions into problem (1), it can be shown that Eq. (11) will generate a linear system for both cases. Generally both linear systems can be stated as

$$-\sum\sum K^*_{i,j,r,s}U_{r,s} = \sum\sum C^*_{i,j,r,s}f_{r,s} \tag{12}$$

where,

$$K^*_{i,j,r,s} = \int_a^b \int_a^b \left(\frac{\partial R_{i,j}}{\partial x}\frac{\partial R_{r,s}}{\partial x}\right)\, dxdy + \int_a^b \int_a^b \left(\frac{\partial R_{i,j}}{\partial y}\frac{\partial R_{r,s}}{\partial y}\right)\, dxdy$$

$$C^*_{i,j,r,s} = \int_a^b \int_a^b \left(R_{i,j}(x,y)R_{r,s}(x,y)\right)\, dxdy$$

Practically, the linear system in Eq. (12) for the full-sweep and half-sweep cases will be easily rewritten in the stencil form respectively as follows:

1. Full-sweep stencil (Zienkiewicz [19], Twizell [15], Fletcher [5])

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} U_{i,j} = \frac{h^2}{12}\begin{bmatrix} 0 & 1 & 1 \\ 1 & 6 & 1 \\ 1 & 1 & 0 \end{bmatrix} f_{i,j} \tag{13}$$

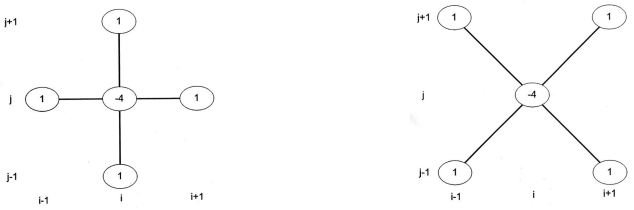
2. Half-sweep stencil

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & -4 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} U_{i,j} = \frac{h^2}{6}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 5 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} f_{i,j}, \quad i = 1 \tag{14}$$

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & -4 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} U_{i,j} = \frac{h^2}{6}\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 6 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} f_{i,j}, \quad i \neq 1, n \tag{15}$$

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & -4 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} U_{i,j} = \frac{h^2}{6}\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 5 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} f_{i,j}, \quad i = n \tag{16}$$

The stencil forms in Eqs. (13) till (16), which are based on the first order triangle finite element approximation equation, can be used to represent the full-sweep and half-sweep computational molecules as shown in Figure 4. Actually, the computational molecules in Figure 4 involve seven node points in formulating their approximation equations. However, two of its coefficients are zero. Apart of this, the form of the computational molecules for both triangle finite element schemes is the same compared to the existing five points finite difference scheme, see Abdullah [1], Abdullah and Ali [2], Yousif and Evans [17].



a) **b)**

Fig. 4. Computational molecules of the Galerkin triangle finite element scheme in case of a). the full-sweep and b). the half-sweep cases.

3 Formulation and implementation of the 4 Point-EDG*SOR*-RB

The 4 Point-EDG iterative method has been introduced by Abdullah [1] in solving problem (1) by using a five point rotated finite difference approximation equation. The iterative implementation of this method will involve only red node points till the converge test is reached. While approximate values of the remaining node points will be computed directly by direct methods. Further discussion of this method based on the finite difference approach can be found in Ibrahim and Abdullah [6], Yousif and Evans [17], Abdullah and Ali [2].

Even though Yousif and Evans [17] have discussed the 4, 6, and 9 Point-EDG iterarive methods based on the finite difference method, this paper will just consider the 4 Point-EDG method, which will be derived by using Eqs. (14) to (16). Let a four point group be considered to form a (4x4) linear system given as (Abdullah [1])

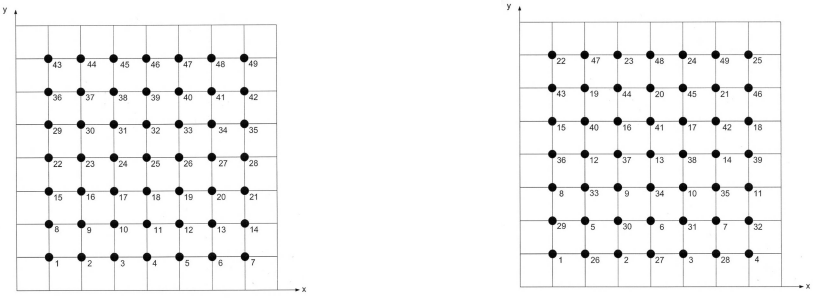
$$\begin{bmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & 0 & 0 \\ 0 & 0 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \\ U_{i+1,j} \\ U_{i,j+1} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \tag{17}$$

where,

$$\begin{aligned} S_1 &= U_{i-1,j-1} + U_{i-1,j+1} + U_{i+1,j-1} - b^*_{i,j}, \\ S_2 &= U_{i,j+2} + U_{i+2,j} + U_{i+2,j+2} - b^*_{i+1,j+1}, \\ S_3 &= U_{i,j-1} + U_{i+2,j-1} + U_{i+2,j+1} - b^*_{i+1,j}, \\ S_4 &= U_{i,j+1} + U_{i-1,j} + U_{i-1,j+2} - b^*_{i,j+1}. \end{aligned}$$

Again this system in Eq. (17) can be rewritten as two (2x2) linear systems, which are independent for each other. The 4 Point-EDG*SOR* can be easily shown as

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k+1)} = (1-w) \begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k)} + \frac{w}{15} \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}, \tag{18}$$



a) **b)**

Fig. 5. a). and b). show the NA and RB ordering strategies for FSGS at $n = 7$

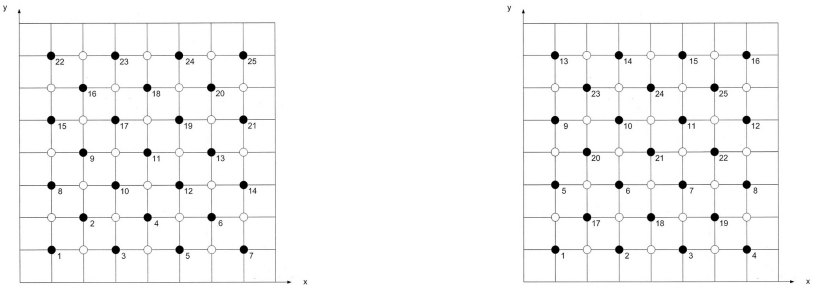
$$\begin{bmatrix} U_{i+1,j} \\ U_{i,j+1} \end{bmatrix}^{(k+1)} = (1-w) \begin{bmatrix} U_{i+1,j} \\ U_{i,j+1} \end{bmatrix}^{(k)} + \frac{w}{15} \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} S_3 \\ S_4 \end{bmatrix}, \tag{19}$$

where the value of w is in range of [1, 2]. If w=1, this method will be called as the existing 4 Point-EDG(GS) method. Apart of the block point iterations, combination of iterative schemes and ordering strategies, which have been proven, can accelerate the convergence rate, see Parter [12], Evans and Yousif [3], Zhang [18]. In this section, however, two ordering strategies will be discussed this paper such as the lexicography (NA) and red-black (RB) and then being applied to the FSGS, HSGS, EDG and EDGSOR iterative methods, called as FSGS-NA, HSGS-NA, HSGS-RB, EDG-RB and EDGSOR-RB methods respectively. In comparison, the FSGS-NA method will be used as a control method and then the HSGS-RB will acts as the control of comparison of numerical results.

It can be seen from Figures 5 and 6 by using the half-sweep triangle finite element approximation equations in Eqs. (14) till (16), the position of numbers in the solution domain for $n = 7$ shows on how both HSGS-NA and HSGS-RB methods will be performed by starting at number 1 and ending at the last number. Similarly, the implementation of the EDG and EDGSOR iterative methods can be shown in Figure 7. However, EDG and EDGSOR iterative methods with RD approach will be conducted in the next section.

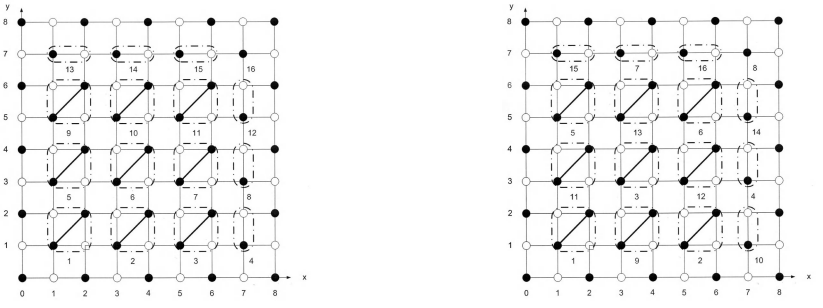
4 Numerical Experiments

To indicate the effectiveness of the 4 Point-EDGSOR-RB scheme by using the half-sweep linear finite element approximation equation in Eqs. (14) till (16) based on the Galerkin scheme, three items will be considered in comparison such as the number of iterations, execution time and maximum absolute error. Some numerical experiments were conducted in solving the following 2D Poisson equation (Abdullah [1])



a) **b)**

Fig. 6. a). and b). show the NA and RB ordering strategies for HSGS at $n = 7$



a) **b)**

Fig. 7. a). and b). show the NA and RB ordering strategies for 4 Point-EDG*SOR* at $n = 7$

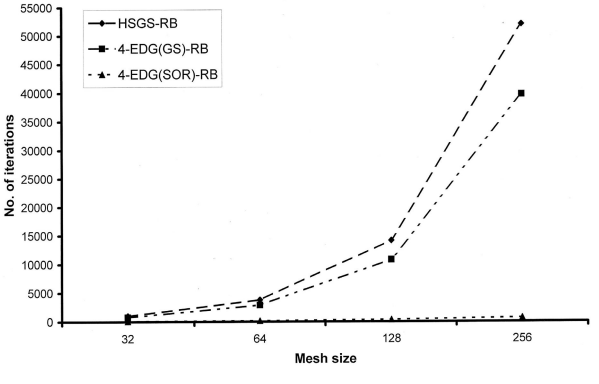


Fig. 8. Number of iterations versus mesh size of HSGS-RB, 4 Point-EDG-RB, and 4 Point-EDG*SOR*-RB

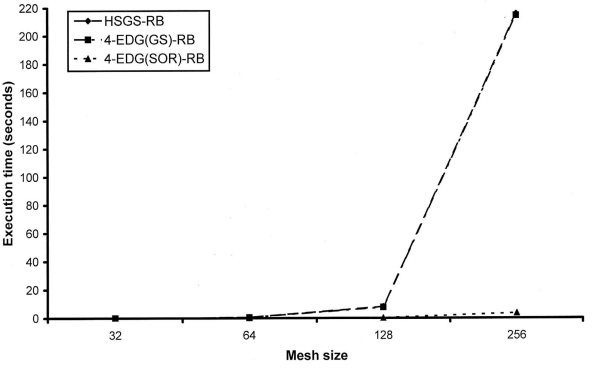


Fig. 9. The execution time (seconds) versus mesh size of HSGS-RB, 4 Point-EDG-RB, and 4 Point-EDG*SOR*-RB

Table 1. Comparison of number of iterations, execution time (in seconds) and maximum errors for the iterative methods

Number of iterations				
Methods	Mesh size			
	32	64	128	256
FSGS-NA	1986	7368	27164	99433
HSGS-NA	1031	3829	14159	52020
HSGS-RB	1027	3825	14152	52008
4-EDG-RB	785	2917	10798	39729
4-EDG*SOR*-RB	89 (w=1.732)	178 (w=1.854)	349 (w=1.924)	690 (w=1.961)
Execution time (seconds)				
Methods	Mesh size			
	32	64	128	256
FSGS-NA	0.14	2.08	30.51	498.89
HSGS-NA	0.03	0.63	9.08	218.74
HSGS-RB	0.03	0.56	8.19	215.70
4-EDG-RB	0.02	0.52	7.75	214.44
4-EDG*SOR*-RB	0.01	0.03	0.25	3.67
Maximum absolute errors				
Methods	Mesh size			
	32	64	128	256
FSGS-NA	1.4770e-4	3.6970e-5	9.3750e-6	2.8971e-6
HSGS-NA	5.7443e-4	1.6312e-4	4.4746e-5	1.1932e-5
HSGS-RB	5.7443e-4	1.6312e-4	4.4746e-5	1.1932e-5
4-EDG-RB	5.7443e-4	1.6312e-4	4.4746e-5	1.1932e-5
4-EDG*SOR*-RB	5.7443e-4	1.6312e-4	4.4746e-5	1.1932e-5

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = \left(x^2 + y^2\right) \exp(xy), \quad (x, y) \in D = [a, b] \times [a, b] \tag{20}$$

Then boundary conditions and the exact solution of the problem (20) are defined by

$$U(x, y) = exp(xy), \quad (x, y) = [a, b] \times [a, b] \tag{21}$$

All results of numerical experiments, obtained from implementation of the FSGS-NA, HSGS-NA, HSGS-RB, 4 Point-EDG-RB, and 4 Point-EDGSOR-RB methods, have been recorded in Table 1. In the implementation mentioned above, the convergence criteria considered the tolerance error, $\epsilon = 10^{-10}$. Figures 8 and 9 show number of iterations and the execution time against mesh respectively.

5 Conclusion

In the second section, it has shown that the full-sweep and half-sweep triangle finite element approximation equations based on the Galerkin scheme can be easily represented in Eqs. (13) till (16). Through numerical results collected in Table 1, the findings in Figures 8 and 9 show that number of iterations have declined approximately $23.56 - 23.73\%$ and $91.33 - 98.07\%$ correspond to the 4 Point-EDG-RB, and 4 Point-EDGSOR-RB methods compared to HSGS-RB method. In fact, the execution time versus mesh size for both 4 Point-EDG-RB, and 4 Point-EDGSOR-RB methods are much faster approximately $0.58 - 33.33\%$ and $66.67 - 98.30\%$ respectively than the HSGS-RB method. Thus, it can be concluded that the 4 Point-EDGSOR-RB method is slightly better than the HSGS-RB and 4 Point-EDG-RB methods. This is due to the computational complexity of the Half-Sweep iterative methods is nearly 50% of the FSGS-NA method. Again, approximate solutions for the HSGS-RB, 4 Point-EDG-RB, and 4 Point-EDGSOR-RB methods are in good agreement compared to the FSGS-NA method. For future works, this study will be continued to investigate on the use of the 4 Point-EDGSOR-RB as a smoother for the halfsweep multigrid (Othman & Abdullah [8,9]) and the development and implementation of the Modified Explicit Group (MEG) (Othman & Abdullah [10], Othman *et al.* [11])and the Quarter-Sweep Iterative Alternating Decomposition Explicit (QSI-ADE) (Sulaiman *et al.* [14]) methods by using finite element approximation equations.

References

1. Abdullah, A.R.: The Four Point Explicit Decoupled Group (EDG) Method: A Fast Poisson Solver. Intern. Journal of Computer Mathematics 38, 61–70 (1991)
2. Abdullah, A.R., Ali, N.H.M.: A comparative study of parallel strategies for the solution of elliptic pde's. Parallel Algorithms and Applications 10, 93–103 (1996)
3. Evan, D.J., Yousif, W.F.: The Explicit Block Relaxation method as a grid smoother in the Multigrid V-cycle scheme. Intern. Journal of Computer Mathematics 34, 71–78 (1990)

4. Fletcher, C.A.J.: The Galerkin method: An introduction. In: Noye, J. (pnyt.). Numerical Simulation of Fluid Motion, pp. 113–170. North-Holland Publishing Company, Amsterdam (1978)
5. Fletcher, C.A.J.: Computational Galerkin method. Springer Series in Computational Physics. Springer, Heidelberg (1984)
6. Ibrahim, A., Abdullah, A.R: Solving the two-dimensional diffusion equation by the four point explicit decoupled group (EDG) iterative method. Intern. Journal of Computer Mathematics 58, 253–256 (1995)
7. Lewis, P.E., Ward, J.P.: The Finite Element Method: Principles and Applications. Addison-Wesley Publishing Company, Reading (1991)
8. Othman, M., Abdullah, A.R.: The Halfsweeps Multigrid Method As A Fast Multigrid Poisson Solver. Intern. Journal of Computer Mathematics 69, 219–229 (1998)
9. Othman, M., Abdullah, A.R.: An Effcient Multigrid Poisson Solver. Intern. Journal of Computer Mathematics 7, 541–553 (1999)
10. Othman, M., Abdullah, A.R.: An Efficient Four Points Modified Explicit Group Poisson Solver. Intern. Journal of Computer Mathematics 76, 203–217 (2000)
11. Othman, M., Abdullah, A.R., Evans, D.J.: A Parallel Four Point Modified Explicit Group Iterative Algorithm on Shared Memory Multiprocessors. Parallel Algorithms and Applications 19(1), 1–9 (2004) (On January 01, 2005 this publication was renamed International Journal of Parallel, Emergent and Distributed Systems)
12. Parter, S.V.: Estimates for Multigrid methods based on Red Black Gauss-Seidel smoothers. Numerical Mathematics 52, 701–723 (1998)
13. Sulaiman, J., Hasan, M.K., Othman, M.: The Half-Sweep Iterative Alternating Decomposition Explicit (HSIADE) method for diffusion equations. In: Zhang, J., He, J.-H., Fu, Y. (eds.) CIS 2004. LNCS, vol. 3314, pp. 57–63. Springer, Heidelberg (2004)
14. Sulaiman, J., Othman, M., Hasan, M.K.: Quarter-Sweep Iterative Alternating Decomposition Explicit algorithm applied to diffusion equations. Intern. Journal of Computer Mathematics 81, 1559–1565 (2004)
15. Twizell, E.H.: Computational methods for partial differential equations. Ellis Horwood Limited, Chichester (1984)
16. Vichnevetsky, R.: Computer Methods for Partial Differential Equations, vol. I. Prentice-Hall, New Jersey (1981)
17. Yousif, W.S., Evans, D.J.: Explicit De-coupled Group iterative methods and their implementations. Parallel Algorithms and Applications 7, 53–71 (1995)
18. Zhang, J.: Acceleration of Five Points Red Black Gauss-Seidel in Multigrid for Poisson Equations. Applied Mathematics and Computation 80(1), 71–78 (1996)
19. Zienkiewicz, O.C.: Why finite elements? In: Gallagher, R.H., Oden, J.T., Taylor, C., Zienkiewicz, O.C. (eds.) Finite Elements In Fluids-Volume, vol. 1, pp. 1–23. John Wiley & Sons, Chichester (1975)

Performance of Particle Swarm Optimization in Scheduling Hybrid Flow-Shops with Multiprocessor Tasks

M. Fikret Ercan[1] and Yu-Fai Fung[2]

[1] School of Electrical and Electronic Engineering, Singapore Polytechnic, 500 Dover Rd., S139651, Singapore
mfercan@sp.edu.sg
[2] Department of Electrical Engineering, Hong Kong Polytechnic University, Hung Hom Kowloon, Hong Kong SAR
eeyffung@inet.polyu.edu.hk

Abstract. In many industrial and computing applications, proper scheduling of tasks can determine the overall efficiency of the system. The algorithm, presented in this paper, tackles the scheduling problem in a multi-layer multiprocessor environment, which exists in many computing and industrial applications. Based on the scheduling terminology, the problem can be defined as multiprocessor task scheduling in hybrid flow-shops. This paper presents a particle swarm optimization algorithm for the solution and reports its performance. The results are compared with other well known meta-heuristic techniques proposed for the solution of the same problem. Our results show that particle swarm optimization has merits in solving multiprocessor task scheduling in a hybrid flow-shop environment.

1 Introduction

Scheduling concerns with allocating limited resources to optimize an objective and it is an immensely studied field of engineering. Multiprocessor task scheduling is a generalized form of classical machine scheduling problem where a task is processed by more than one processor [2]. Multiprocessor task scheduling is a challenging problem encountered in computer and manufacturing processes and it is difficult to solve even in its simplest form [3, 6]. Earlier studies in multiprocessor task scheduling are mainly concerned with a single stage setting of the processor environment. However, there are many practical problems that require jobs to go through more than one stage. This problem, known as multistage flow-shop, is also vastly studied in scheduling context though most of these studies consider single processor at each stage [13]. With the advances made in technology, in many practical flow-shop environments, we encounter parallel processors at each stage instead of single processor (see for instance [1, 4, 11, 12, 17]). This scheduling problem is defined as multiprocessor task scheduling in a hybrid flow-shop environment and it is gaining attention from the research community. As the complexity of the problem increases with the increasing number of layers, the early studies tackled a simple form of the problem, two-layer

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 309 – 318, 2007.
© Springer-Verlag Berlin Heidelberg 2007

flow-shops with multiprocessors [14, 8]. Simple list based heuristics as well as meta-heuristics such as Simulated Annealing (SA), Tabu Search (TS) and Genetic Algorithm (GA) are introduced for the solution. A comparative performance study of these heuristics can be found in [5]. However, there are many practical applications where multiprocessor tasks are executed in a setting that includes more than two stages. Studies tackling this problem using meta-heuristics are quite recent. Such meta-heuristic solutions reported in literature include GA [16], TS [15], SA [18] and recently Ant Colony System [19]. Since solution to this problem finds applications in various fields in industry and in computing, finding a good solution is important. In this paper, Particle Swarm Optimization (PSO) approach for the solution is presented and results are compared with other algorithms published in the literature.

The PSO is another evolutionary algorithm [9, 10] gaining popularity within the research community. It mimics the behavior of flying birds and their communication mechanism to solve optimization problems. It is based on a constructive cooperation between particles, in contrast with survival of the fittest approach used in other evolutionary methods [9]. It has many advantages which made it worth to experiment and study its performance for the scheduling problem presented in this paper. The algorithm is simple, fast and easy to code. It is not computationally intensive in terms of memory requirements and time. Furthermore, it has only a few parameters to tune.

The problem considered in this paper is formulated as follows: There is a set J of n independent and simultaneously available Multi-Processor Tasks (MPTs) to be processed in a multi-stage flow-shop, where stage j consists of m_j identical parallel processors $(j = 1,2,..,k)$. Each $MPT_i \in J$ should be processed on $p_{i,j}$ identical processors simultaneously at stage j without interruption for a period of $t_{i,j}$ $(i = 1,2,...,n \text{ and } j = 1,2,...,k)$. Hence, each $MPT_i \in J$ is characterized by its processing time, $t_{i,j}$, and its processor requirement, $p_{i,j}$ $(i = 1,2,...,n \text{ and } j = 1,2,...,k)$. Tasks flow from one stage to another by utilizing any of the processors while satisfying the flow-shop and the MPT constraints. In the remainder of this paper, set of MPTs that are going to be processed on a multi stage system will be called as jobs and the scheduling problem is basically finding a sequence of jobs that can be processed on the system in the shortest possible time. The following assumptions are made when modeling the problem:

- All the processors are continuously available from time 0 onwards and each processor can handle no more than one MPT at a time.
- The processing time and number of processors required at each stage are known in advance.
- Set-up times and inter-processor communication time are all included in the processing time and it is independent of the job sequence.

In the following section, we describe the design of the algorithm. Next, the computational study is presented. We then report and discuss the computational results. Conclusions are given in the last section.

2 The PSO Algorithm

The PSO algorithm is initialized with a population of random solutions which is similar to evolutionary algorithms. Each individual solution flies in the problem space with a velocity that is adjusted depending on the experiences of an individual and the population. There are various models of PSO. In this study, the global model is employed and it is defined as:

$$V_{id} = WV_{id} + C_1 R_1(P_{id} - X_{id}) + C_2 R_2(P_{gd} - X_{id}) \text{ and } X_{id} = X_{id} + V_{id}.$$

In the above equation, V_{id} is the velocity of particle i and it represents the distance traveled from its current position. W is the inertia weight. X_{id} represents the particle position. P_{id} is the local best solution (also called "*pbest*") and P_{gd} is the global best solution (also called "*qutgbest*"). C_1 and C_2 are acceleration constants which drive particles towards the local and the global best positions. R_1 and R_2 are two random numbers within the range $\{0,1\}$. The algorithm follows the following steps:

```
Initialize swarm with random positions and velocities;
  begin
    repeat
    For each particle evaluate the fitness i.e.
makespan of the schedule;
    if current fitness of a particle is better than Pid
then set Pid to the current value;
    if Pid is better than the global best then set Pgd to
current particle fitness value;
    Change the velocity and position of the particle;
    until termination = True
end.
```

The initial swarm and particle velocity are generated randomly. Each particle consists of a sequence of job numbers representing the n number of jobs on a machine with k number of layers where each layer has m_j identical processors $(j = 1,2,..,k)$. The fitness of a particle is measured with maximum completion time of all jobs. A function is developed to map a given job sequence to the multiprocessor system and compute the maximum completion time as shown in Figure 1. A particle with lowest completion time is a good solution and has to be kept in search process.

In the example given in Figure 1, it is assumed that a job sequence is generated as $S_1 = \{2,3,1,4,5\}$ and a machine is made up of two stages where each stage contains four identical processors. At stage 1, jobs are iteratively allocated to processors from the list starting from time 0 onwards. As job 2 is the first in the list, it is scheduled at time 0. It is important to note that although there are enough available processors to schedule job 1 at time 0 this will violate the precedence relationship established in the list. Therefore, job 1 is scheduled to time instance 3 together with job 3 and this does not violate the precedence relationship given in list S_1. Once all the jobs are

scheduled at first stage, a new list is produced for the succeeding stage based on completion of jobs at previous stage and precedence relationship given in list S_1. In the new list, $S_2 = \{2,1,3,4,5\}$, job 1 is scheduled before job 3 since it is available earlier than job 3. At time instance 7, jobs 3, 4 and 5 are all available. Job 3 is scheduled first since its completion time is earlier at stage 1. Although, there is enough processor to schedule job 5 at time 8 this will again violate the order given in list S_1, hence it is scheduled together with job 4.

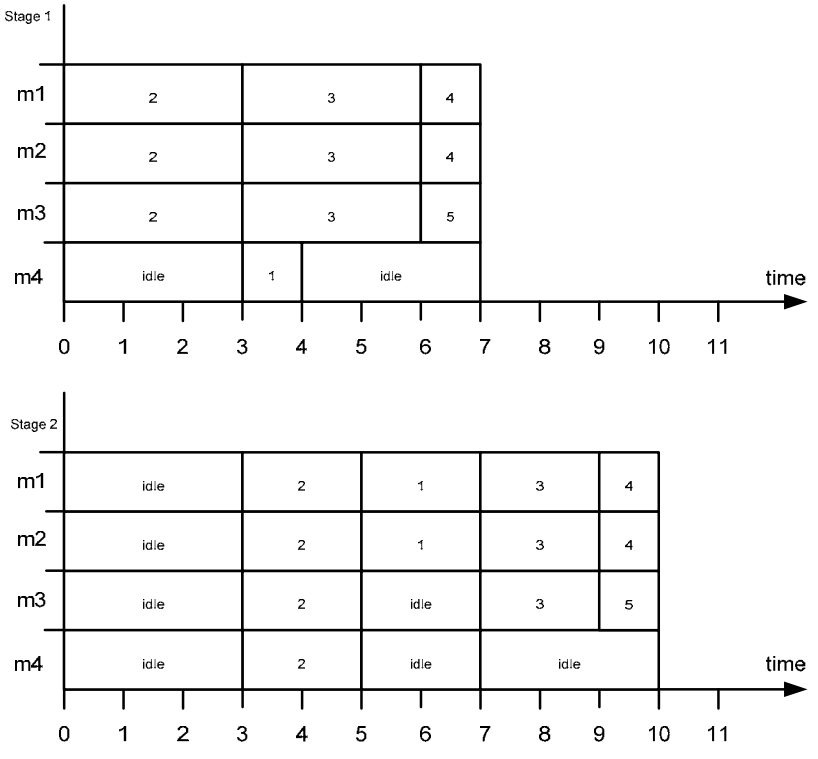


Fig. 1. The schedule of job sequence [2, 3, 1, 4, 5] after being decoded to processors with scheduling algorithm. Idle periods of the processors are labeled as idle.

The setting of parameters for PSO are based on our empirical studies as well as studies and experiences of other researchers. The acceleration constants C_1 and C_2 are set to 2.0 and initial population of swarm is set to 100. Inertia weight, W, determines the search behavior of the algorithm as a large value facilitates searching new locations whereas a small value provides a finer search in the current area. A balance can be established between global and local exploration by decreasing the inertia weight during the execution of the algorithm. This way PSO tends to have a

more global search ability at the beginning and more local search ability towards the end of the execution. In our PSO algorithm, an exponential function is used to set the inertia weight which is defined as:

$$W = W_{end} + (W_{start} - W_{end})e^{\frac{x\alpha}{x_{\max}}}$$

where, W_{start} is the starting and W_{end} is the ending inertia values. W_{start} and W_{end} are set as 1.5 and 0.3 respectively. In addition, x shows current iteration number and $x_{\max}$ maximum iteration number which is set to 10000 iterations. An integer constant α is used to manipulate the gradient of the exponentially decreasing W and it is set to 4.

In our application, X_{id} and V_{id} are used to generate and modify solutions therefore they are rounded off to the nearest integer and limited to a maximum value of n which is the maximum number of jobs. That is, position coordinates are translated into job sequence in our algorithm and a move in the search space is obtained by modifying the job sequence.

3 Computational Results

An intensive computational study is conveyed to examine the performance of the PSO algorithm in minimizing the overall completion time of all jobs. Furthermore, the effects of parameters such as number of jobs and processor configurations on the performance of the algorithm are also investigated. The algorithm is implemented with the Java language and run on a PC, with 2GHz Pentium processor and 1024 MB memory. The results are presented in terms of Average Percentage Deviation (*APD*) of the solution from the lower bound which is expressed as $APD = \frac{C_{\max} - LB}{LB} \times 100$. Here, $C_{\max}$ indicates the completion time of the job and LB indicates the lower bound calculated for the problem instance. The lower bound used in this study was developed by Oğuz et. al. [15] and it is given with the following formula:

$$LB = \left\{ \max_{i \in M} \left\{ \min_{j \in J} \sum_{l=1}^{i-1} t_{l,j} \right\} + \frac{1}{m_i} \sum_{j \in J} p_{i,j} \times t_{i,j} + \min_{j \in J} \left\{ \sum_{l=i+1}^{l} t_{l,j} \right\} \right\}$$

In the above formula, M and J represent the set of stages and set of jobs respectively. We used the benchmark data available at Oğuz's web-site (http://home.ku.edu.tr/~coguz/). Data set contains instances for two types of processor configurations: (i) random number of processors in each stage selected from a set of {1,..., 5} and (ii) identical number of processors at each stage which is fixed to 5 processors. For both configurations, a set of 10 problem instances is randomly

produced for various number of jobs (n=5, 10, 20, 50, 100) and various number of stages (k=2, 5, 8). For each n and k value, the average APD is evaluated based on 10 problem instances. Tables 1 and 2 present the APD results obtained for PSO. Furthermore, we have compared the results of PSO algorithm with GA [16], TS [15] and ACS [19]. The efficiency of GA presented in [16] is closely related to control parameters, as well as cross over and mutation techniques. The authors introduce a new cross over technique named NXO, which aimed to capture the characteristics of the problem. In addition, they implemented the Partially Matched Crossover, PMX [7] for comparison and employed two types of mutation techniques namely Insertion mutation (iM) and Swap mutation (sM). In Tables 1 and 2, the best results reported by GA are listed. However, for the sake of completeness, crossover and mutation techniques used for which the best results were obtained are also indicated. Similarly, TS [15] and ACS [19] results obtained for the same problem sets are listed in the tables.

From the experimental studies, it can be observed that PSO outperformed the TS algorithm and approximated the results of GA and ACS algorithms. It outperformed TS in 10 problems out of a total of 12 for the random processor case and the best improvement rate was 82.13% when k=5 and n=100. Similarly, it outperformed 8 out of 12 problems for the fixed processor case though the best improvement rate was 60.09% when k=8 and n=10. When compared to another swarm based algorithm ACS, the results were inferior. ACS outperformed more than 50% in 6 of the 12 problems for the random processors case. On the other hand, the best improvement for ACS was 31.8% for the fixed processor case. PSO outperformed GA 4 in 12 problems for random processors with best improvement of 54% and 1 in 12 problems for fixed processors. It can be observed that GA is superior as compared to other heuristics methods when the number of stages is small.

For a given number of stages, increasing the number of jobs results in a decrease in *APD* value. This characteristic is observed for all the results presented in Tables 1 and 2. Apparently, the lower bound becomes more effective and close to the optimal solution. On the other hand, for a given number of jobs, increasing the number of stages results in an increase in *APD* as the lower bound becomes less effective. From the published results, it is observed that when the number of processors are fixed, that is $m_i = 5$, the scheduling problem becomes more difficult to solve and *APD* results are relatively higher. The same characteristic is also observed for PSO algorithm.

We also experimented the PSO algorithm on a set of data obtained from an actual machine vision system. This data can also be obtained from the web-site (http://home.ku.edu.tr/~coguz/). In this experiment, we compared the convergence of the algorithms. It can be seen from Figure 2, PSO converges much faster than TS and GA. In addition, PSO finds a slightly better solution for 12 job problem. All three algorithms demonstrate a fast convergence. In fact, search termination rule can be limited to 1000 iterations to complete scheduling in an acceptable time interval which is more suitable for practical applications.

CPU time spend for the experiments is another indicator of the algorithm performance though it won't be a fair comparison as different processors and compilers were used for each reported algorithm. However, in order to have a rough idea on the speed

performance of PSO, the observations for one easy and one difficult problem case are collated in Table 3. Termination criterion is 10000 iterations for all the algorithms. It can be seen from Table 3 that PSO is approximately 48% to 35% faster than reported GA CPU timings and 55% to 57% faster than TS timings [16].

Table 1. *APD* of the algorithms for 10 random instances. Processors setting is $m_j \sim [1,5]$.

k	*n*	GA	*TS*	*ACS*	*PSO*
2	10	1.60 (NXO/iM)	3.00	1.60	2.7
2	20	0.80(NXO/iM)	2.88	1.92	2.88
2	50	0.69 (NXO/iM)	2.23	2.37	2.38
2	100	0.35 (NXO/iM)	9.07	0.91	1.82
5	10	11.89 (PMX/iM	29.42	9.51	10.33
5	20	5.54 (NXO/iM)	24.40	3.07	8.6
5	50	5.11 (NXO/iM)	10.51	1.51	3.31
5	100	3.06 (NXO/iM)	11.81	1.05	2.11
8	10	16.14 (NXO/sM)	46.53	16.50	18.23
8	20	7.98 (NXO/iM)	42.47	6.77	12.03
8	50	6.03 (NXO/iM)	11.81	2.59	5.98
8	100	4.12 (PMX/iM)	21.50	1.33	8.78

Table 2. *APD* of the algorithms for 10 random instances. Processors setting is $m_j = 5$.

k	*N*	GA	*TS*	*ACS*	*PSO*
2	10	6.13 (NXO/iM)	10.82	12.62	13.8
2	20	7.10 (NXO/iM)	7.25	10.73	10.75
2	50	3.34 (NXO/iM)	5.80	8.17	10.32
2	100	2.87 (NXO/iM)	5.19	5.66	7.43
5	10	11.32 (NXO/sM)	45.14	26.09	29.6
5	20	10.78(NXO/iM)	35.13	15.11	19.4
5	50	14.91 (NXO/iM)	28.64	13.11	14.17
5	100	11.02 (NXO/iM)	26.49	12.45	12.45
8	10	25.98 (NXO/iM)	77.21	25.14	30.81
8	20	24.13(NXO/iM)	62.99	25.18	26.74
8	50	21.87 (NXO/iM)	54.25	22.23	27.01
8	100	19.46 (NXO/iM)	36.05	13.90	20.39

Table 3. Average CPU time (in seconds) spend by TS, GA and PSO. Processors setting is $m_j = 5$.

k	*N*	GA	*TS*	*PSO*
2	10	12.14	14.11	6.3
8	100	2109.9	3181.32	1388.67

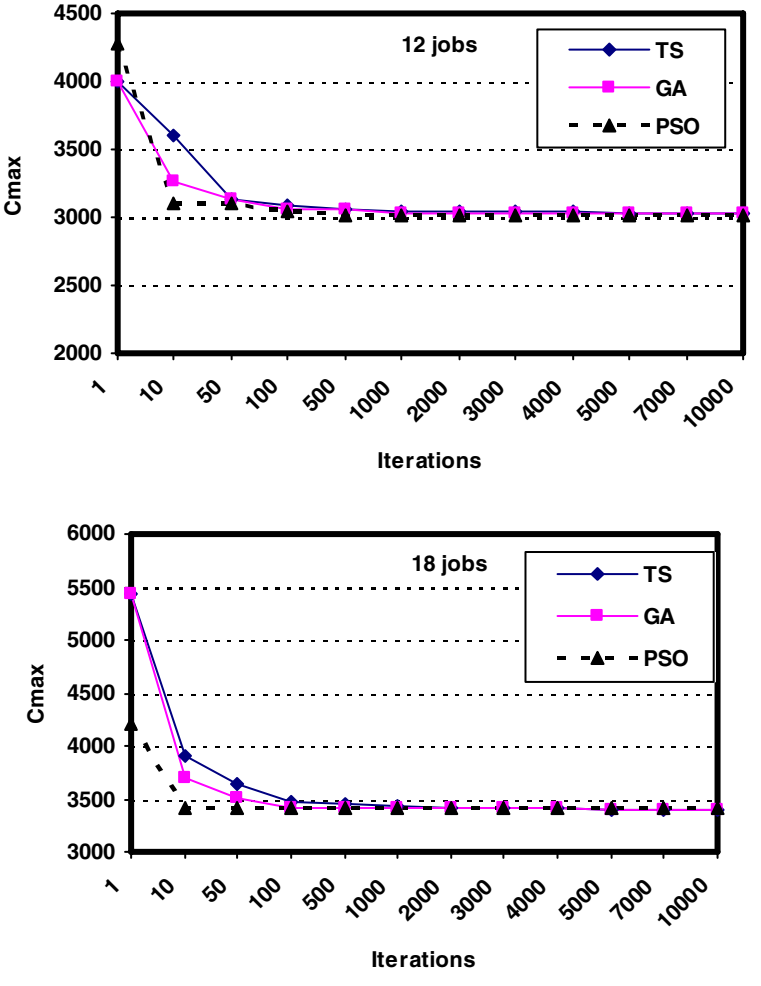


Fig. 2. Convergence of TS, GA and PSO algorithms for a machine vision system

4 Summary

In this paper, performance of a PSO algorithm for scheduling hybrid flow-shops with multiprocessors is presented. A job is made up of interrelated multiprocessor tasks where a multiprocessor task is modeled with its processing requirements and processing time. The performance of the PSO algorithm on its capacity to minimize maximum completion time of all jobs is evaluated. The algorithm outperforms the TS and approximates the performance of GA and ACS algorithms. However, considering that the simple form of the PSO algorithm is employed in this study, as compared to hybrid techniques employed in ACS [19] or problem specific cross over technique used

in GA [16], results produced by PSO is satisfactory. In addition, in terms of convergence and CPU time, we observe that PSO outperforms TS and GA. Apparently, hybrid methods will improve the performance of PSO significantly and it is planned for our future study.

Acknowledgement

The first author gratefully acknowledges the contribution of his final year project students in coding and testing of the algorithm.

References

1. Caraffa, V., Ianes, S., Bagchi, T.P., Sriskandarajah, C.: Minimizing Make-Span in Blocking Flow-Shop Using Genetic Algorithms. International Journal of Production Economics 70, 101–115 (2001)
2. Chan, J., Lee, C.Y.: General Multiprocessor Task Scheduling. Naval Research Logistics 46, 57–74 (1999)
3. Drozdowski, M.: Scheduling Multiprocessor Tasks - An Overview. European Journal of Operational Research 94, 215–230 (1996)
4. Ercan, M.F., Fung, Y.F.: The Design and Evaluation of a Multiprocessor System for Computer Vision. Microprocessors and Microsystems 24, 365–377 (2000)
5. Ercan, M.F., Oğuz, C.P: Performance of Local Search Heuristics on Scheduling a Class of Pipelined Multiprocessor Tasks. Computers and Electrical Engineering 31, 537–555 (2005)
6. Garey, E.L., Johnson, D.S., Sethi, R.: The Complexity of Flow-shop and Job-shop Scheduling. Math. Operations Research 1, 117–129 (1976)
7. Goldberg, D., Lingle, R.: Alleles, Loci, and the Traveling Salesman Problem. In: Proceedings of the First International Conference on Genetic Algorithms and Their Applications, pp. 154–159 (1985)
8. Gupta, J.N.D, Hariri, A.M.A., Potts, C.N: Schedules for a Two-stage Hybrid Flow-shop with Parallel Machines at First Stage. Ann. Oper. Res. Soc. 69, 171–191 (1997)
9. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE Int. Conf. on Neural Network, pp. 1942–1948 (1995)
10. Kennedy, J.: The Particle Swarm: Social Adaptation of Knowledge. In: Proceedings of IEEE Int. Conf. on Evolutionary Computation, pp. 303–308 (1997)
11. Krawczyk, H., Kubale, M.: An Approximation Algorithm for Diagnostic Test Scheduling in Multi-computer Systems. IEEE Trans. Computers 34(9), 869–872 (1985)
12. Lee, C.Y., Cai, X.: Scheduling One and Two-processors Tasks on Two Parallel Processors. IIE Transactions 31, 445–455 (1999)
13. Linn, R., Zhang, W.: Hybrid Flow-Shop Schedule: A Survey. Computers and Industrial Engineering. 37, 57–61 (1999)
14. Oğuz, C., Ercan, M.F., Cheng, T.C.E., Fung, Y.F.: Heuristic Algorithms for Multiprocessor Task Scheduling in a Two Stage Hybrid Flow Shop. European Journal of Operations Research 149, 390–403 (2003)
15. Oğuz, C., Zinder, Y., Do, V., Janiak, A., Lichtenstein, M.: Hybrid Flow-Shop Scheduling Problems with Multiprocessor Task Systems. European Journal of Operations Research 152, 115–131 (2004)

16. Oğuz, C., Ercan, M.F.: A Genetic Algorithm for Hybrid Flow-Shop Scheduling with Multiprocessor Tasks. Journal of Scheduling 8, 323–351 (2005)
17. Scala, M.L., Bose, A., Tylavsky, J., Chai, J.S.: A Highly Parallel Method for Transient Stability Analysis. IEEE Transactions on Power Systems 5, 1439–1446 (1990)
18. Sivrikaya-Serifoglu, F., Tiryaki, I.U.: Multiprocessor Task Scheduling in Multistage Hybrid Flow-Shops: A Simulated Annealing Approach. In: Proceedings of 2nd Int. Conf. on Responsive Manufacturing, pp. 270–274 (2002)
19. Ying, K.C, Lin, S.W.: Multiprocessor Task Scheduling in Multistage Hybrid Flow-Shops: an Ant Colony System Approach. International Journal of Production Research 44, 3161–3177 (2006)

Branch-and-Bound Algorithm for Anycast Flow Assignment in Connection-Oriented Networks

Krzysztof Walkowiak

Chair of Systems and Computer Networks, Faculty of Electronics,
Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
Krzysztof.Walkowiak@pwr.wroc.pl

Abstract. Our discussion in this article centers around the problem of anycast flow assignment. *Anycast* is a *one-to-one-of-many* delivery technique that allows a client to choose a content server of a set of replicated servers. We formulate an optimization problem of anycast flow assignment in a connection-oriented network, which is a binary and NP-complete problem. Therefore, to enable calculation of optimal solution we propose an exact algorithm based on the branch-and-bound approach. There has not been any study we are aware of that proposes and examines the branch-and-bound algorithm for anycast flow optimization. We describe the most important elements of the branch-and-bound algorithm and present results of numerical experiments.

Keywords: anycast, branch-and-bound, connection-oriented network.

1 Introduction

Most of transmissions issued in the current Internet are *unicast* – two individual systems connect *one-to-one* to transmit data. However, in order to overcome problems encountered in computer networks (e.g. network latency, congestion, growing costs) new techniques must be developed. One of the most interesting approaches that can make possible the development of new services in the Internet is the *anycast* paradigm. Anycast is a *one-to-one-of-many* technique, in which one of connection's endpoints must be selected among many possible nodes. Anycast transmission can be applied to develop many new services, e.g. distribution of electronic music, movies, books; software updates, etc.

In this work we address the problem of anycast flow assignment in connection-oriented (c-o) networks. Examples of network techniques using the c-o transmission are MPLS (Multiprotocol Label Switching), ATM (Asynchronous Transfer Mode), DWDM (Dense Wavelength Division Multiplexing) [7], [11]. Connection-oriented techniques enable traffic engineering and QoS performance in carrier networks. Therefore, there are widely applied in backbone networks.

In this paper we consider an existing backbone network using a connection-oriented technique. In many cases the network is in an operational phase and augmenting of its resources (links, capacity, replica servers) or changing location of replica servers is not possible in a short time perspective. Therefore, the best method

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 319 – 329, 2007.
© Springer-Verlag Berlin Heidelberg 2007

to improve network performance to an existing backbone transport network is such a routing of existing demands that guarantees allocation of flows minimizing the network delay.

The problem can be formulated as follows

Given	network topology, traffic demand pattern, location of replica servers, link capacity
Minimize	network delay
Over	selection of replica server, routing (path assignment)
Subject to	connection-oriented flow constraints, capacity constraints

The presented problem is NP-complete. This follows from the fact that the unicast version of non-bifurcated flow assignment is NP-complete [11]. Since we introduce additional constraint of anycast flow, our problem is also NP-complete. In previous works of the author we solved the considered problem by a simple heuristic algorithm based on the Flow Deviation method [13] and a heuristic algorithm applying Lagrangean relaxation and subgradient optimization [15]. Since the considered problem is NP-complete in this work we propose an exact algorithm based on the branch-and-bound approach. Having optimal solution we can estimate the quality of results yielded by previously developed heuristics.

2 Anycasting in Connection-Oriented Networks

Anycast is a *one-to-one-of-many* technique to deliver a packet to one of many hosts. Comparing to unicast transmission, in anycasting one of the endpoints of the transmission must be selected among many possible nodes. One of the most well-known technologies that applies anycast traffic is Content Delivery Network (CDN). CDN is defined as mechanisms to deliver a range of content to end users on behalf of the origin Web servers. Information is offloaded from source sites to other content servers located in different locations in the network. For each request, CDN tries to find the closest server offering the requested Web page. CDN delivers the content from the origin server to the replicas that are much closer to end-users. The set of content stored in CDNs servers is selected carefully. Thus, the CDNs' servers can approach the hit ratio of 100%. It means that almost all request to servers are satisfied [14]. Other applications of anycasting in computer networks are Domain Name Service, Web Service, Distributed Database System, host auto-configuration, overlay network, peer-to-peer (P2P) systems, sensor networks [1-3], [5-6], [8], [10]..

Most of previous work on anycasting concentrates on IP networks using connection-less transmission modeled as bifurcated multicommodity flows [3], [5], [8], [10]. Since we consider a connection-oriented network (e.g. MPLS), we model the network flow as non-bifurcated multicommodity flow. Consequently, the anycast request (demand) must consist of two connections: one from the client to the server (upstream) and the second one in the opposite direction (downstream). Upstream connection sends user's requests. Downstream connection carries the requested data. Thus, each anycast demand is defined by a following triple: client node, upstream bandwidth requirement and downstream bandwidth requirement. In contrast, a unicast demand is

defined by a following triple: origin node, destination node and bandwidth requirement. To establish a unicast demand a path satisfying requested bandwidth and connecting origin and destination nodes must be found. Optimization of anycast demands is more complex. The first step is the server selection process. Next, when the server node is selected, both paths: upstream and downstream can be calculated analogously to unicast approach. However, the main constraint is that both connections associated with a particular anycast demand must connect the same pair of network nodes and one of these nodes must host a server.

For more information on modeling anycast flows in connection-oriented networks please refer to [13-16].

3 Anycast Flow Assignment Problem

To mathematically represent the problem, we introduce the following notations:

V	Set of n vertices representing the network nodes.
A	Set of m arcs representing network directed links.
P	Set of q connections in the network. A connection can be of two types: downstream or upstream.
π_p^k	A candidate route of connection p with index k. In the case of an upstream connection the route connects a client node and a node hosting a content server. Analogously, downstream candidate path connects a server and a client node.
Π_p	The index set of candidate routes (paths) π_p^k of connection p.
x_p^k	A binary decision variable, which is 1 if route $k \in \Pi_p$ is selected in connection p and 0 otherwise.
X	Set of variables x_p^k, which are equal to one. X is called a selection.
c_a	Capacity of arc a.
Q_p	Bandwidth requirement of connection p.
$\tau(p)$	Index of the connection associated with the same anycast demand as connection p. If p is a downstream connection $\tau(p)$ must be an upstream connection and vice versa.
δ_{pa}^k	Binary variable, which is 1 if arc a belongs to route π_p^k and is 0 otherwise.
$s(\pi)$	Server node of route π. If π is a route of a downstream connection, $s(\pi)$ is the origin node of the route. If π is a route of an upstream connection, $s(\pi)$ is the destination node of route π.
T	total arrive rate of messages in the network

The anycast non-bifurcated flow assignment (AFA) problem is as follows

$$\min_X DEL(X) = \frac{1}{T} \sum_{a \in A} \frac{f_a}{(c_a - f_a)} \tag{1}$$

subject to

$$f_a = \sum_{p \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k Q_p \quad \forall a \in A \tag{2}$$

$$\sum_{k \in \Pi_p} x_p^k = 1 \qquad \forall p \in P \tag{3}$$

$$f_a \le c_a \qquad \forall a \in A \tag{4}$$

$$\sum_{k \in \Pi_p} x_p^k s(\pi_p^k) = \sum_{k \in \Pi_{\tau(p)}} x_{\tau(p)}^k s(\pi_{\tau(p)}^k) \qquad \forall p \in P \tag{5}$$

$$x_p^k \in \{0,1\} \qquad \forall p \in P; k \in \Pi_p \tag{6}$$

$$X = \left(\bigcup_{p,k:x_p^k=1} \{x_p^k\} \right) \tag{7}$$

The goal is to minimize the function of network delay [4], [7], [11] .(2) is a definition of a flow that is transmitted on arc *a*. Condition (3) states that each connection can use only one route. (4) is a capacity constraint, i.e. flow of each arc cannot exceed its capacity. Constraint (4) follows from the technical limitations of computer networks – each network link has a limited capacity. Equation (5) guarantees that two routes associated with the same anycast demand connect the same pair of nodes. Constraint (6) ensures that decision variables are binary ones. Note that the constraints' set doesn't include a constraint guaranteeing that upstream (downstream) connection ends (starts) in a node with a replica server. This follows from the construct of set Π_p that contains only feasible candidate routes, i.e. each route connects the client node and a replica node. (7) is a definition of the selection that includes all decision variables x_p^k, which are equal to 1. The selection *X* enables calculation of the aggregate flow of each arc according to (2).

Notice that if we remove constraint (5) connected with anycast flows we will obtain the single-path allocation problem, which according to [11] is NP-complete. Therefore, the AFA problem given by (1)-(7) also is NP-complete.

4 Calculation Scheme

In the proposed branch-and-bound algorithm we start with a feasible initial solution X_1 and generate a sequence of selections X_r. In order to obtain selection X_1 we solve the AFA problem using heuristic algorithm proposed in [15]. Each new selection X_s is obtained from a certain selection X_r of the sequence by complementing a normal variable x_p^k by a reverse variable x_p^i in the following way $X_s = \left(X_r - \{x_p^k\}\right) \cup \{x_p^i\}$. Both variables must be associated with the same connection *p* in order to satisfy the condition (3). The generating process can be represented as a branch-and-bound decision

tree. Each node of the decision tree represents a selection. Each arc of the tree represents a pair of selections (X_r, X_s) such that X_s is obtained from X_r. We say that the selection X_s is a successor of the selection X_r if there is a path from X_r to X_s. For every selection X_r we constantly fix a subset $U_r \in X_r$ and momentarily fix a set U_r^t. The variables in U_r are constantly fixed and denote the path from the initial selection X_1 to the current selection X_r in the decision tree. Each momentarily fixed variable in U_r^t is the variable abandoned during the backtracking process. There are two important elements of the branch-and-bound algorithm that are calculated for each selection X_r: lower bound of the criterion function and branching rules. The lower bound is calculated to check if a "better" solution may be found. If the testing is negative we abandon the considered selection X_r and backtrack to the selection X_j from which the selection X_r was generated. The basic task of the branching rules is to find the variables for complementing to generate a new selection with the least possible value of the criterion function. There are three types of branching rules: correction operation, regulation operation and choice operation. The first one – correction – is applied if constraint (5) is not satisfied. The regulation operation is used if the arc capacity constraint (4) does not hold. Finally, if all constraints of the considered problem are satisfied in the current selection X_r, we apply the choice operation.

5 Lower Bound

Notice that the objective function (1) of the AFA problem (1-7) is strictly convex. The FD (Flow Deviation) algorithm proposed in [4] can solve optimally the bifurcated mutlicommodity flow problem with a strictly convex objective function. Therefore, in problem (1-7) we relax constraint (6) to obtain a bifurcated mutlicommodity anycast flow problem as follows

$$0 \le x_p^k \le 1 \qquad \forall k \in \Pi_p, p \in P \tag{8}$$

Next, we develop a modified version of the FD algorithm – called AFD – accordingly to the anycast flow defined by (5). The modification concentrates on calculation of the shortest route flow. The FD algorithm calculates a shortest route of each commodity according to the metric, which is a partial derivative of the objective function. In our approach we find a pair of routes of associated connections (commodities) p and $\tau(p)$, which are shorter than any other pair of routes and connect the same pair of nodes due to constraint (5). The AFD algorithm tries to deviate a part of the current flow to new shortest routes in order to minimize the objective function. Repeating of this process guarantees that the algorithm converges to the global optimum [4].

6 Branching Rules

The elementary task of the branching rule is to choose the normal variable and reverse variable for complementing and generating a successor X_s of the current selection X_r with the least possible criterion function. If the current selection X_r is feasible, we apply the choice operation. Otherwise, we use either the correction operation if the

constraint (5) does not hold in X_r or the regulation operation if the arc capacity constraint (4) is not satisfied in flows calculated accordingly to selection X_r. Note that choice and regulation operations are applied in branch-and-bound algorithms proposed in the context of unicast flow problems [7], [12]. The correction operation is a new element of the branch-and-bound algorithm developed as a result of the anycast flow defined by (5). Let us define the following function

$$\Delta DEL_{pr}^{ki} = Q_p\left(\sum_{a\in A}\delta_{pa}^{k} l_a^{DEL}(X_r) - \sum_{a\in A}\delta_{pa}^{i} l_a^{DEL}(X_r)\right) \tag{9}$$

where $l_a^{DEL}(X_r)$ denotes the partial derivative of the function *DEL* (1) calculated for selection X_r. It is proved in [7] that if $X_s = \left(X_r - \{x_p^k\}\right)\cup\{x_p^i\}$ then

$$DEL(X_s) \geq DEL(X_r) - \Delta DEL_{pr}^{ki} \tag{10}$$

Let $E_r = X_r - U_r$ be a set that contains all variables x_p^k equal to 1 and not constantly fixed. Set M_r includes all variables x_p^k equal to 0 that can be selected as reverse variables in the choice operation

$$\begin{aligned} M_r = &\left(\bigcup_{p:x_p^k\in E_r, x_{\tau(p)}^j\notin U_r}\{x_p^i : i\in \Pi_p\}\right)\cup \\ &\left(\bigcup_{p:x_p^k\in E_r, x_{\tau(p)}^j\in U_r}\{x_p^i : i\in \Pi_p^{s_r(\tau(p))}\}\right) - E_r \end{aligned} \tag{11}$$

where $\Pi_p^{s_r(\tau(p))}$ is a set of routes of connection p, which have the same server node as connection $\tau(p)$, where $x_{\tau(p)}^j \in U_r$. In the choice operation we select variables $x_p^k \in E_r$ and $x_p^i \in \left(M_r - U_r^t\right)$, where the value of ΔDEL_{pr}^{ki} given by (9) is maximized and generate a new selection $X_s = \left(X_r - \{x_p^k\}\right)\cup\{x_p^i\}$. Due to (10) such a procedure assures that the objective function *DEL* will be reduced at least by ΔDEL_{pr}^{ki}.

The regulation operation is executed if the current selection X_r does not satisfy the capacity constraint (4). This occurs if there exists arc a for which $f_a > c_a$. The main goal of this operation is to reduce the flow on arc a. Let E_{ra} include all variables $x_p^k \in E_r$ associated with routes that include arc a

$$E_{ra} = \bigcup_{p:x_p^k\in E_r, \delta_{pa}^k=1}\{x_p^k\} \tag{12}$$

Let M_{ra} be a set including variables associated with routes not containing arc a that can be selected in complementing

$$M_{ra} = \left(\bigcup_{p: x_p^k \in E_r, x_{\tau(p)}^j \notin U_r} \{ x_p^i : i \in \Pi_p \quad \text{and} \quad \delta_{pa}^i = 0 \} \right) \cup \left(\bigcup_{p: x_p^k \in E_r, x_{\tau(p)}^j \in U_r} \{ x_p^i : i \in \Pi_p^{s_r(\tau(p))} \quad \text{and} \quad \delta_{pa}^i = 0 \} \right) \tag{13}$$

In the regulation operation we choose variables $x_p^k \in E_{ra}$ and $x_p^i \in \left(M_{ra} - U_r^t \right)$, for which the value of ΔDEL_{pr}^{ki} given by (9) is maximized. The new selection is calculated as $X_s = \left(X_r - \{ x_p^k \} \right) \cup \{ x_p^i \}$.

Now we discuss the correction operation. Let assume that the constraint (5) does not hold in the case of connection p, where $x_p^k \in X_r$. Consequently, we must construct a new selection X_s by changing the decision variable of connection $\tau(p)$. Let $s_r(p)$ denote the server node of connection p given by variable $x_p^k \in X_r$. Set N_{rp} contains all variables that can be selected in complementing

$$N_{rp} = \bigcup_{i \in \Pi_{\tau(p)}} \{ x_{\tau(p)}^i : s(\pi_{\tau(p)}^i) = s_r(p) \} \tag{14}$$

In the correction operation the new selection is generated as $X_s = \left(X_r - \{ x_{\tau(p)}^j \} \right) \cup \{ x_{\tau(p)}^i \}$, where $x_{\tau(p)}^j \in X_r$. Variable $x_{\tau(p)}^i \in \left(N_{rp} - U_r^t \right)$ is selected to maximize value of (9).

7 Results

We implemented the above presented branch-and-bound algorithm in C++ and performed simulations on a network consisting of 14 nodes and 54 arcs. 10 various demand patterns were tested with 9 different server locations including 2, 3 and 4 servers. Since we consider an existing backbone network using a connection-oriented technique, we assume that the servers are already located. However, to examine performance of the branch-and-bound algorithm and heuristics we test various locations of servers.

The goal of numerical experiments was twofold. First, we wanted to examine the performance of the branch-and-bound algorithm in terms of the decision time and the size of the solution tree in various cases of server selection and demand patterns. Second, we planed to evaluate the effectiveness of heuristic algorithms proposed in [13], [15] comparing results of these algorithms against optimal results yielded by the branch-and-bound algorithm.

Fig. 1 presents the number of nodes in the solution tree of branch-and-bound algorithm as a function of the aggregate bandwidth requirement. Each curve in the graph represents a specific replica server set size. We can watch that the size of the solution tree generally changes with the number of servers and traffic pattern. This follows

from the construct of the algorithm. Size of the solution tree and consequently the decision time of the algorithm depends on the branching rules and the lower bound. The branching rule is responsible to generate a new selection. The lower bound enables the algorithm to reduce the number of analyzed selections. On Fig. 2 we report the decision time in the same cases as in Fig. 1.

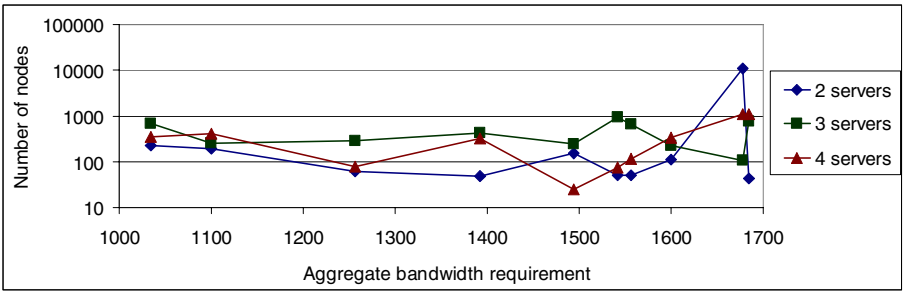


Fig. 1. Number of nodes in the solution tree of the branch-and-bound algorithm as a function of the aggregate bandwidth requirement and the number of servers

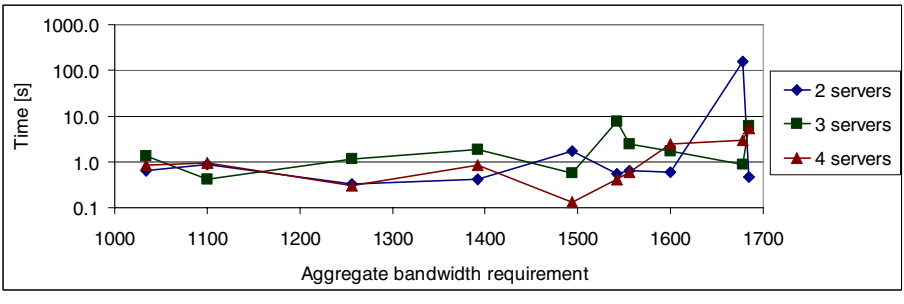


Fig. 2. Decision time of the branch-and-bound algorithm as a function of the aggregate bandwidth requirement and the number of servers

If we compare the number of nodes (Fig. 1) against the decision time (Fig. 2) we can notice that there is a general correlation – more nodes in the decision tree means more time spent to find the exact solution. However, a detailed analysis shows that the amount of time spent on one node of the decision tree grows with the value of aggregate bandwidth requirement irrespective of the number of servers. The most time consuming operation of the branch-and-bound algorithm is the lower bound. Recall that to calculate the lower bound we apply the AFD algorithm based on the Flow Deviation method [4]. The number of iterations of the algorithm and consequently the execution time of the AFD algorithm is not deterministic. Our results suggests that the execution time of algorithm AFD grows with the network congestion. The basic observation is that the number of servers and traffic pattern strongly affect the branch-and-bound algorithm performance in terms of the decision time and the size of the solution tree.

The main novelty of the branch-and-bound algorithm proposed in this work - comparing to previous works on nonbifurcated flows (e.g. [7], [9]) - is a new branching rule called correction, which is a consequence of the anycast flow constraint (5). To evaluate the correction operation Fig. 3 shows the percentage number of correction operations in the overall number of branching rules. We can watch that on average about 5% of all branching rules is the correction operation. However, curves plotted with various number of servers shows that in some cases the number of correction operations grows to 20%.

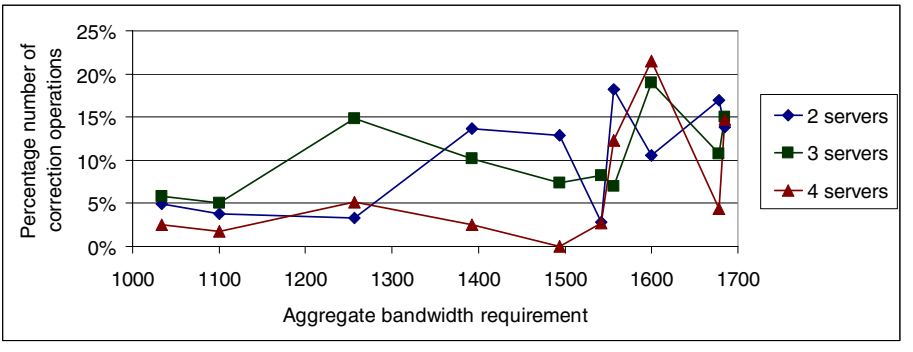


Fig. 3. The percentage number of correction operations as a function of the aggregate bandwidth requirement and the number of servers

Since the branch-and-bound algorithm developed in this paper yields optimal solutions, we can evaluate quality of results of heuristic algorithms proposed in previous works to solve the AFA problem. In Table 1 we report results of two heuristic algorithms ANBFD [13] and LRH [15] compared to optimal results. In columns 3 and 4 we report the average gap between heuristics and exact algorithm. In columns 5 and 6 we present the number of tests, in which the heuristic yields the optimal result. We can notice that the LRH algorithm based on the Lagrangean relaxation produces result close to optimal ones.

Table 1. Results of heuristics ANBFD and LRH against optimal results

Number of servers	Number of tests	Average gap to optimal		Number of optimal results	
		ANBFD	LRH	ANBFD	LRH
2	40	3.23%	0.15%	1	25
3	40	1.86%	0.01%	3	32
4	10	0.89%	0.00%	2	10

In Fig. 4 we present curves of the network delay function obtained with 2, 3 and 4 servers as a function of the aggregate bandwidth requirement of all anycast demands. The general trend is the value of delay function increases with the network load and decreases as the number of servers grows. We can notice that introducing of new

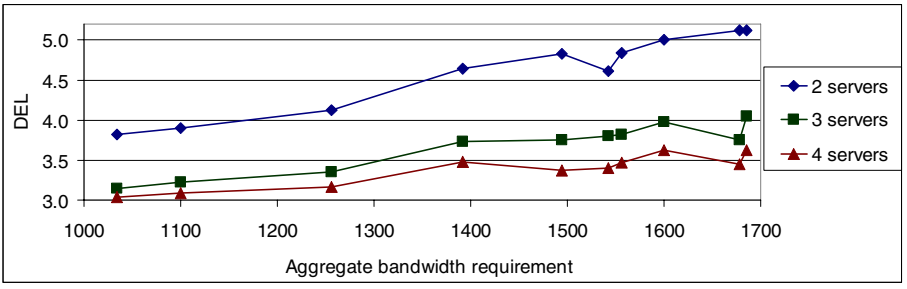


Fig. 4. Optimal values of average delay function

servers can substantially reduce the network delay. Summarizing all tests, the value of the delay function yielded with 4 servers is 7.75% and 26.40% lower than the corresponding value obtained with 3 servers and 2 servers, respectively.

8 Conclusions

In this paper, a new branch-and-bound algorithm to solve the anycast flow assignment problem in connection-oriented networks has been proposed. To our best survey, this is the first study that applies the branch-and-bound algorithm to the anycast flow problem. We have focused on the two most important elements of the algorithm: lower bound and branching rules. Main novelties of these operations follow from anycast flow constraints. Next, we have run simulations to examine performance of the algorithm for various scenarios of replica server location and the traffic demand pattern. In particular, the analysis shows that performance expressed by the decision time and size of the solution tree depends strongly on the number of servers and traffic demand pattern. Moreover, having optimal solution we could evaluate heuristics proposed in previous works and the conclusion is that the heuristics based on the Lagrangean relaxation method [15] approaches optimal results. Finally, optimal results yielded by the branch-and-bound algorithm confirms our previous observations [15] that introducing more servers to the network substantially reduces the network delay.

Acknowledgements. This work was supported by a research project of the Polish State Committee for Scientific Research carried out in years 2005-2007.

References

1. Awerbuch, B., Brinkmann, A., Scheideler, C.: Anycasting in adversarial systems: routing and admission control. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 1153–1168. Springer, Heidelberg (2003)
2. Ballani, H., Francis, P.: Towards a Global IP Anycast Service. In: Proc. SIGCOMM'05, Philadelphia, pp. 301–312 (2005)
3. Doi, S., Ata, S., Kitamura, K., Murata, M.: IPv6 anycast for simple and effective service-oriented communications. IEEE Communication Magazine 5, 163–171 (2004)

4. Fratta, L., Gerla, M., Kleinrock, L.: The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design. Networks 3, 97–133 (1973)
5. Hao, F., Zegura, E., Ammar, M.: QoS routing for anycast communications: motivation and an architecture for DiffServ networks. IEEE Communication Magazine 6, 48–56 (2002)
6. Hou, Y.T., Shi, Y., Sherali, H.D.: Optimal base station selection for anycast routing in wireless sensor networks. IEEE Transactions on Vehicular Technology 3, 813–821 (2006)
7. Kasprzak, A.: Designing of Wide Area Networks. Wroclaw Univ. of Tech. Press (2001)
8. Lin, C., Lo, J., Kuo, S.: Load-Balanced Anycast Routing. In: Proc, of the Parallel and Distributed Systems. In: Tenth international Conference on (Icpads'04), Washington (2004)
9. Markowski, M., Kasprzak, A.: The web replica allocation and topology assignment problem in wide area networks: algorithms and computational results. In: Gervasi, O., Gavrilova, M., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3483, pp. 772–781. Springer, Heidelberg (2005)
10. Peng, G.: CDN: Content Distribution Network. sunysb.edu/tr/rpe13.ps.gz (2003)
11. Pióro, M., Medhi, D.: Routing, Flow, and Capacity Design in Communication and Computer Networks. Morgan Kaufman Publishers, Seattle, Washington (2004)
12. Walkowiak, K.: A Branch-and-bound Algorithm for Primary Routes Assignment in SurvivableConnection Oriented Networks. Computational Optimization and Applications 2, 149–171 (2004)
13. Walkowiak, K.: Heuristic Algorithm for Anycast Flow Assignment in Connection-Oriented Networks. In: Sunderam, V.S., van Albada, G.D., Sloot, P.M.A., Dongarra, J.J. (eds.) ICCS 2005. LNCS, vol. 3516, pp. 1092–1095. Springer, Heidelberg (2005)
14. Walkowiak, K.: A Unified Approach to Survivability of Connection-Oriented Networks. In: Yolum, p., Güngör, T., Gürgen, F., Özturan, C. (eds.) ISCIS 2005. LNCS, vol. 3733, pp. 3–12. Springer, Heidelberg (2005)
15. Walkowiak, K.: Lagrangean Heuristic for Anycast Flow Assignment in Connection-Oriented Networks. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J.J. (eds.) ICCS 2006. LNCS, vol. 3991, pp. 618–625. Springer, Heidelberg (2006)
16. Walkowiak, K.: Survivable Routing of Unicast and Anycast Flows in MPLS Networks. In: Proc. 3rd EURO-NGI Conference on Next Generation Internet Networks, pp. 72–79 (2007)

Quasi-hierarchical Evolutionary Algorithm for Flow Optimization in Survivable MPLS Networks

Michał Przewoźniczek[1] and Krzysztof Walkowiak[2]

[1] Faculty of Computer Science and Management, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
Michal.Przewozniczek@pwr.wroc.pl
[2] Chair of Systems and Computer Networks, Faculty of Electronics, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
Krzysztof.Walkowiak@pwr.wroc.pl

Abstract. In this paper we address the problem of working paths optimization in survivable MPLS network. We focus on an existing facility network, in which only network flows can be optimized to provide network survivability using the local repair strategy. The main goal of our work is to develop an effective evolutionary algorithm (EA) for considered optimization problem. The novelty is that the proposed algorithm consists of two levels. The "high" level applies typical EA operators. The "low" level idea is based on the hierarchical algorithm idea. However, the presented approach is not a classical hierarchical algorithm. Therefore, we call the algorithm quasi-hierarchical. We present a precise description of the algorithm and results of simulations run over various networks.

Keywords: evolutionary algorithm, survivability, MPLS.

1 Introduction

Recently, we can observe a fast development of computer networks, which implies that the survivability of communication service is becoming a fundamental problem. Connection-oriented network technologies like MultiProtocol Label Switching (MPLS) use quite simple and effective method to enable network survivability. Each connection - called label switched path (LSP) - has a working path (route) and a recovery (backup) path (route). The working route is used for transmitting of data in normal, failure-free state of the network. After a failure of the working path, the failed connection is switched to the backup route. There are two major configurations of working and recovery paths in MPLS [10]: local repair (LR) and global repair (GR). In the former method the backup path is established around the failed element (link or node) and the node immediately upstream of the fault is the one to start recovery. In the latter method the reaction to a failure takes place from an end-to-end viewpoint for each connection. For more details on survivable MPLS networks refer to [4], [10].

In this paper we concentrate on an existing facility network, i.e. we do not consider capacity planning and topological design. Our goal is to find working routes for connections in order to prepare the network for future network failures. We focus on local

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 330 – 342, 2007.
© Springer-Verlag Berlin Heidelberg 2007

repair method, however our approach can be also applied for global repair. We use an objective function –called LFL (Lost Flow in Link), which was already successfully applied to survivable assignment of network flows [11]. Since according to [5] the considered problem is NP-complete, we develop an effective heuristic algorithm based on evolutionary algorithm called HEFAN (**H**ierarchical **E**volutionary algorithm for **F**low **A**ssigment in **N**on-bifurcated commodity flow). Comparing to our previous work on this subject [8-9], based on results of simulations and our analysis, we modified main elements of the HEFAN algorithm including crossing and mutation procedures and added the low level fitness function.

2 Problem Formulation

Our objective in this section is twofold. First, we want to define an objective function - called LFL (Lost Flow in Link) - that shows the preparation of the network to the restoration process. Next, we will formulate an optimization problem using LFL function. To mathematically represent the problem we introduce the following notations.

Sets:

- V set of n vertices representing the network nodes.
- A set of m arcs representing network directed links.
- P set of q connections in the network.
- Π_p the index set of candidate working paths (routes) for connection p.
- X_r set (selection) of variables x_p^k, which are equal to one. X_r determines the unique set of currently selected working paths

Indices:

- p connections (demands) in the network, used as subscript
- k candidate routes, used as superscript
- a arcs (directed links), used as subscript
- r selections, used as subscript

Constants:

- δ_{pa}^k equal to1, if arc a belongs to path k realizing connection p; 0 otherwise
- Q_p volume (estimated bandwidth requirement) of connection p
- c_a capacity of arc a

Variables:

- x_p^k decision variable, which is 1 if the working route $k \in \Pi_p$ is selected for connection p and 0 otherwise
- f_a flow of arc a
- g_v^{out} aggregate flow of outgoing arcs of v; $g_v^{\text{out}} = \sum_{i:o(i)=v} f_i$
- e_v^{out} aggregate capacity of outgoing arcs of v; $e_v^{\text{out}} = \sum_{i:o(i)=v} c_i$

We denote by $o: A \to V$ and $d: A \to V$ functions defining the origin and destination node of each arc. For each $a \in A$ we call $\text{in}(a) = \{k \in A \mid d(k) = d(a), k \neq a\}$ the set of incoming arcs of $d(a)$ except a, and $\text{out}(a) = \{k \in A \mid o(k) = o(a), k \neq a\}$ the set of outgoing arcs of $o(a)$ except a.

Definition 1. The global non-bifurcated m.c. flow denoted by $\underline{f} = [f_1, f_2, ..., f_m]$ is defined as a vector of flows in all arcs. We call a flow $\underline{f}$ feasible if for every arc $a \in A$ the following inequality holds

$$\forall a \in A: f_a \leq c_a \tag{1}$$

Inequality (1) ensures that in every arc flow is not greater then capacity. This inequality is called a capacity constraint.

For the sake of simplicity we introduce the following function

$$\varepsilon(x) = \begin{cases} 0 & \text{for} \quad x \leq 0 \\ x & \text{for} \quad x > 0 \end{cases} \tag{2}$$

To analyze properties of the local repair we consider a failure of arc $k \in A$. In the local repair flow on the arc k must be rerouted by the origin node of k. Therefore, residual capacity of arcs outgoing $o(k)$ except k is a potential bottleneck of the restoration process. Notice that if

$$f_k \leq \sum_{i \in \text{out}(k)} (c_i - f_i) \tag{3}$$

then flow of the failed k can be restored using residual capacity of other links leaving the origin node of k. Otherwise if

$$f_k > \sum_{i \in \text{out}(k)} (c_i - f_i) \tag{4}$$

then some flow of the failed link k cannot be restored because residual capacity of other arcs leaving the origin node of k is too small. It means that those arcs block the 100% restoration and some flow of k is lost. Recalling definition of $g^{\text{out}}_{o(k)}$ and $e^{\text{out}}_{o(k)}$ and applying formulas (3-4) we define the function LA^{out}_k of the arc k flow lost in the node $o(k)$ in the following way

$$LA^{\text{out}}_k(\underline{f}) = \varepsilon\left(g^{\text{out}}_{o(k)} - (e^{\text{out}}_{o(k)} - c_k)\right) \tag{4}$$

Notice that in definition (4) the function of lost flow for arc k depends on the whole flow leaving the origin node of k. Therefore, we define the following function of lost flow for all arcs leaving node v

$$LN^{\text{out}}_v(\underline{f}) = \sum_{a:o(a)=v} \varepsilon\left(g^{\text{out}}_v - (e^{\text{out}}_v - c_a)\right) = \sum_{a:o(a)=v} LA^{\text{out}}_a(\underline{f}) \tag{5}$$

Analogously we can define function $LN_v^{\text{in}}(\underline{f})$ that shows how much flow is lost in arcs incoming node *v*. Our goal is to define a function that shows the preparation of the network to local repair. Therefore, we formulate the following function

$$LFL(\underline{f}) = \sum_{v \in V} \left(LN_v^{\text{out}}(\underline{f}) + LN_v^{\text{in}}(\underline{f}) \right) / 2 \tag{6}$$

For more details on LFL function please refer to [11]. Applying function LFL we formulate the optimization problem WP_LFL of working paths assignment as follows

$$\min_{\underline{f}} \quad LFL(\underline{f}) \tag{7}$$

subject to

$$\sum_{k \in \Pi_p} x_p^k = 1 \qquad \forall p \in P \tag{8}$$

$$x_p^k \in \{0,1\} \quad \forall p \in P, \forall k \in \Pi_p \tag{9}$$

$$f_a = \sum_{p \in P} \sum_{k \in \Pi_p} \delta_{pa}^k x_p^k Q_p \quad \forall a \in A \tag{10}$$

$$f_a \le c_a \quad \forall a \in A \tag{11}$$

Condition (8) guarantees that the each connection can use only working route. Constraint (9) ensures that decision variables are binary ones. (10) is a definition of a link flow. Finally, (11) denotes the link capacity constraint.

Problem WP_LFL given by (7-11) is a 0/1 NP problem with linear constraints but it is generally considered as a very hard problem [5]. The main reason why this occurs is that the solution space that includes all possible paths for each connection is extremely large. Size of the problem is large even for relatively small networks. A common approach to solve 0-1 problems is the branch-and-bound (B&B) method. According to [5], application of B&B methods for single path (non-bifurcated) flow allocation is limited to at most medium-size networks (of 20 nodes), and, therefore heuristic methods are of interest. An heuristic algorithm called FDP method was introduced in [11]. Another method to solve the WP_LFL problem is Lagrangean relaxation. The algorithm proposed in [3] – called in our work LRH (Lagrangean Relaxation Heuristic) – solves a problem related to WP_LFL. The only difference is that authors of [3] consider the network delay as the objective function. Therefore, the LRH algorithm can be adapted to the WP_LFL problem.

3 Algorithm Description

In this section we present briefly the various versions of HEFAN (**H**ierarchical **E**volutionary algorithm for **F**low **A**ssigment in **N**on-bifurcated commodity flow) algorithm developed for the WP_LFL problem. It should be noticed that HEFAN

(version 1.0) was successfully applied to a network congestion problem and version 1.0 is the only published HEFAN version so far.

In the literature we can find EA or hybrid-EA algorithms developed for the flow assignment problem, but they are not so advanced as our algorithm. Frequently used notation is a typical EA integer-string [1-2], [6-7] usually representing link weights for different routing protocols [2], [7]. Specified operators are not applied – only crossover and mutation are used. The other difference is that in presented results, HEFAN successfully deals with very long individuals (up to 1260 genes), which is not common for other algorithms. Moreover, previous papers present results conducted for rather small networks (less then 20 nodes) or for much smaller number of flows to route (only in [2] results for the network with over 18000 demands are presented, but all other of 13 cases are below 1100 demands) than presented here. We believe that by adding the "low" level of the algorithm, we achieve two goals: a better search effectiveness and a possibility of using the algorithm for finding the solutions in much bigger search spaces than those searched by other algorithms.

The main objective of the HEFAN algorithm is to find a feasible and the best in terms of fitness function, assignment of paths (routes) for all connections in a connection-oriented network. However, the algorithm needs a proper collection of routes, for which the search will be done. Such a collection can not be neither too narrow nor too wide. Too narrow collection means that the search of the solution space will be limited and there is large probability that this limited space will not include the best solution. On the other hand, too wide set of routes will make the problem intractable due to the tremendous number of possible combinations. Due to these reasons, the HEFAN algorithm is conceptually divided into three abstract parts: the "high" and "low" algorithm level and the routes database. The "high" level works as a typical EA and contains a population of problem solutions (one individual is the set of route proposals for all connections in the network). The "low" level is an intelligent procedure of searching for new routes proposals (one individual is a single route between two nodes, represented by a list of node identifiers) using most of the EA features (crossover, mutation, using fitness function in parent choose process), but not being an EA in fact (no population size constraint – if the route between nodes X,Y has been added to the routes database, it can never be removed from it).

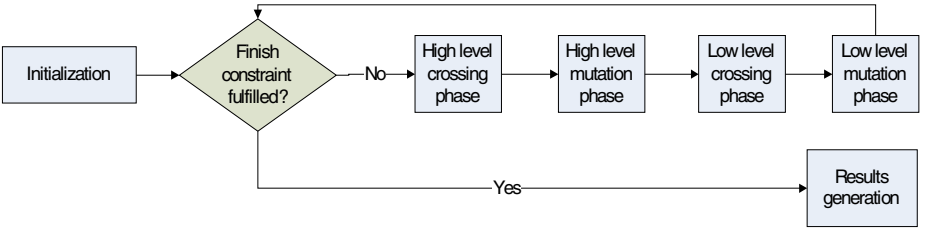


Fig. 1. The HEFAN algorithm versions overview

As shown on Fig. 1 the HEFAN consists of four main phases. Two of them work on the "high" level and two on "low" level. The stopping condition may depend on user demands – it may be a typical time constraint, the maximum generation number,

or the fitness value of the best found solution. In this work we use only the time constraint. In this paper we present the HEFAN version 2.2. The HEFAN version 1.0 was presented in [8-9]. The main modification and a step forward of the version 2.2 when compared with already published version 1.0 is the use of low level fitness function.

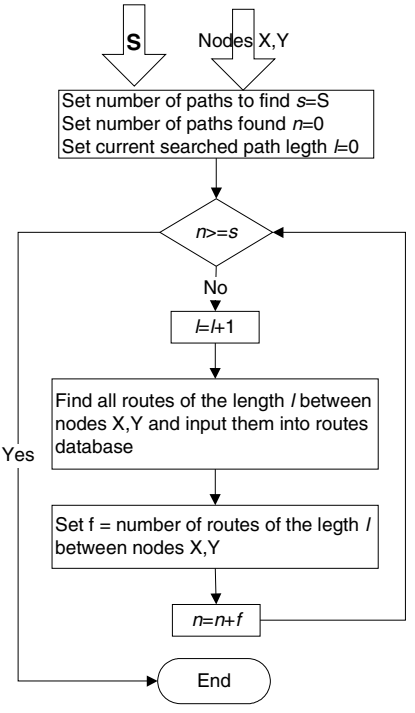


Fig. 2. Searching *s* shortest routes between nodes X,Y procedure

In the initialization phase the input parameters of algorithm (e.g. probabilities of high and low level crossing and mutation) are set. Also the start feasible solution (if any) is generated. For this task we can apply the FDP algorithm [11] or LRH algorithm [3]. The routes database is initiated in the following way. The initial set of routes is built using all paths of the 1, 2, 3 and 4 hops. Additionally the initial routes set includes at least *s* number of shortest routes between all possible node pairs. The procedure of finding at least *s* routes is shown on Fig. 2. For instance, if we want to find with an initialization procedure, at least 1 shortest route between nodes *X* and *Y*, it means that all shortest routes between nodes *X* and *Y* will be added to the database. In this work whenever we say that the algorithm was initiated with at least *s* shortest routes we mean that HEFAN's routes database was initiated with at least *s* shortest routes between all possible node pairs using the algorithm described above.

The high level crossing phase contains typical EA operators and actions. At the end of this phase the new population is generated. The selection is based on the roulette wheel method. Two crossover operators were supported: single-point crossover and

uniform crossover. The decision which one of them is to be used is made randomly at every crossing operation according to the proportions set by the user.

The fitness function used for the experiments is based on the *LFL* function. First we define the following penalty function associated with the link capacity constraint

$$PEN_v(\underline{f}) = \sum_{a:o(a)=v} \varepsilon(f_a - c_a) \tag{12}$$

Next we formulate functions $LN'(\underline{f})$ $LFL'(\underline{f})$, which include the capacity constraint as a penalty function in the following way

$$LN'_v(\underline{f}) = \begin{cases} \dfrac{LN_v^{\text{in}}(\underline{f}) + LN_v^{\text{out}}(\underline{f})}{2} & \text{for} \quad PEN_v(\underline{f}) = 0 \\ \left(\dfrac{LN_v^{\text{in}}(\underline{f}) + LN_v^{\text{out}}(\underline{f})}{2} + PEN_v(\underline{f}) * 10\right)^2 & \text{for} \quad PEN_v(\underline{f}) > 0 \end{cases} \tag{13}$$

$$LFL'(\underline{f}) = \sum_{v \in V} LN'_v(\underline{f}) \tag{14}$$

Finally, the fitness function $F(p)$, where p is the single chromosome, is as follows

$$F(p) = \frac{1}{LFL'(p) + 1} \tag{15}$$

Mutation replaces randomly picked up gene by a different proposition of such gene from routes database (route between the same two nodes as for the mutated gene). The replacing gene is chosen randomly with an equal probability for every proper route in the routes database.

For the low level crossing phase we define the low level fitness function $L(k)$ of the route k. The function $L(k)$ is calculated according to the flow vector given by high level individual p (set of route proposals for all connections in the network) but without the route for the source and destination node of route k.

$$L(k) = \frac{1}{\sum_{p \in P} \sum_{a \in A} \delta_{pa}^k x_p^k L'(k)} \tag{16}$$

where

$$L'(k) = \begin{cases} 1/(c_a - f_a + 1)^2 & \text{for} \quad c_a \geq f_a \\ (f_a - c_a + 1) & \text{for} \quad c_a < f_a \end{cases} \tag{17}$$

In the low level crossing phase the single individual is represented by a single route between the two nodes (represented as the list of nodes the route is going through). Each gene of such an individual represents a node identifier. It is also important to notice that if any route is created during the two low level phases it is checked for containing empty loops (for example route A, B, C, D, C, where A, B, C, D are node identifiers, contains one empty loop and after removing it route will look like this: A, B, C), which are always removed during the route creation process.

In the low level crossing phase the parent selection goes as follows: one high level individual is chosen (using **high** level fitness function and roulette wheel method) for one low level crossing operation. The chosen high level individual is added to the new high level population which is being created during the low level crossing phase. For this high level individual we randomly choose (with equal probability for all) one gene and we compute the low level fitness function $L(k)$ (16) for all routes in the routes database which have the same source and destination node as the chosen gene and we choose two of them using the roulette wheel method. If the chance decides that low level crossing operator is to be used the two selected low level parents are first randomly divided into two pieces. The first part from the first parent is glued with the second part of the second parent for the first offspring, the second offspring is made in an analogical way. If the two pieces do not fit to each other they are glued with the route between the glued nodes, chosen from the routes database using roulette wheel method and low level fitness function $L(k)$ (16). Both low level offspring individuals are added to the routes database and one of them replaces the proper gene in the chosen high level individual (the choice is made randomly, but is proportional to the offsprings $L(k)$ function value).

In the low level mutation phase – like in the low level crossing – a single individual is represented by a single route between two nodes. The procedure of low level mutation goes as follows. For every high level individual the chance of low level mutation is checked, if the low level mutation operator is to be used then one of the high level individual genes is randomly picked up. In the selected gene, which is low level individual randomly chosen part is thrown out and the remaining two other parts are glued with randomly chosen route from routes database (exactly in the same way as it is done in low level crossing operation, but with equal probability for all proper routes – no fitness function is used here).

4 Results

HEFAN was coded in C++ and tested on a family of networks having irregular mesh topology consisting of a master network and a series of progressively sparser networks derived from the master network. Moreover, a regular topology of a mesh was analyzed. We assume that each network arc has capacity of 4800 bandwidth units to model the OC-48 link. Table 1 summarizes the parameters of all sample networks.

Table 1. Parameters of tested networks

Name of network	**104**	**114**	**128**	**144**	**162**
Number of nodes	36	36	36	36	36
Number of links	104	114	128	144	162
Node degree (minimum)	2	2	3	3	3
Node degree (maximum)	5	5	6	6	6
Topology	irregular mesh				
Demand patterns	10	10	10	10	10

In computation tests, it is assumed that there is a requirement to set up a connection for each direction of every node pair. Thus, the total number of demands (commodities) for a 36-node network is 1260. All connections have the same bandwidth requirement in a given demand pattern. In the last row we report the number of traffic demand patterns tested for each experiment.

The tuning of the algorithm was based on testing of 32 different algorithm parameters combinations (high level crossing and mutation probability, low level crossing and mutation probability, proportion between high level single point and uniform crossing operator usage) for chosen 5 different types of networks. All tuning operations were conducted on AMD 3800+ 64 2GB RAM machine. The tuning fixed parameters were population size (200 individuals) and time (60 seconds for one run).

We run two kinds of experiments. In the first one, HEFAN was not using any initial solution. In the second experiment HEFAN was initialized with a solution yielded by LRH algorithm [3] – consequently, HEFAN was used to improve the solution of LRH. Both kinds of main experiments were conducted on the same machine, the time restriction was 360 seconds when HEFAN was initiated with LRH solution and 720 in the opposite case. The routes database was always initiated with all existing routes of length 4 and smaller. Table 2 summarizes all simulated algorithms's configurations. Configuration number 0 depicts the LRH algorithm [3], which was used for reference. Configurations 1 and 4 denote the standard EA (with and without initial solution, respectively), which was also used for reference. We assume that this is a typical EA algorithm which consists only on HEFAN's high level operators. Other configurations represent HEFAN 1.0 and HEFAN 2.2 also with and without starting solution.

Table 2. Tested algorithms' configurations

Configuration. number	Algorithm version	Tuned parameters: high level crossing / high level mutation / high level uniform crossing probability / low level crossing / low level mutation	Init solution	Computation time [s]	Initiated with at least *s* shortest routes
0	LRH	N/A	N/A	N/A	N/A
1	Standard EA	0.3 / 0.2 / 0.5 / N/A / N/A	Yes	360	4
2	HEFAN 1.0	0.3 / 0.1 / 0.0 / 0.5 / 0.4	Yes	360	4
3	HEFAN 2.2	0.3 / 0.2 / 0.3 / 0.5 / 0.4	Yes	360	4
4	Standard EA	0.5 / 0.2 / 0.5 / N/A / N/A	No	720	4
5	HEFAN 1.0	0.5 / 0.1 / 0.0 / 0.5 / 0.4	No	720	4
6	HEFAN 2.2	0.5 / 0.2 / 0.5 / 0.5 / 0.4	No	720	4

To compare results we apply the *competitive ratio* performance indicator, which indicates how well algorithm performs for a particular network. The *competitive ratio* is defined as the difference between result obtained for a considered algorithm and the minimum value of LFL yielded by the best algorithm. For instance, if the HEFAN 1.0 yields 500, and the best obtained result for the same simulation case is 200; the competitive ratio of HEFAN 1.0 is calculated as follows: (500-200)/500=60%. The competitive ratio indicates the quality of algorithm – lower value indicates that the algorithm obtains lower values, closer to the best found solution. For presentation of aggregate results we apply the *average competitive ratio*, which is the average value of competitive ratios for selected tests. The results of average competitive ratio for

different networks for the first experiment (when HEFAN is not initiated with feasible solution) are reported in Fig. 3. Each curve on the graph represents the performance yielded by one of algorithm's configurations. According to Table 2, we show performance of LRH (configuration 0) and standard EA without init solution (configuration 4), HEFAN 1.0 without init solution (configuration 5) and HEFAN 2.2 without init solution (configuration 6). Corresponding results of the second experiment (LRH solution as an init solution) are shown in Fig. 4. According to Table 2, we report average competitive ration of configurations 0, 1, 2 and 3.

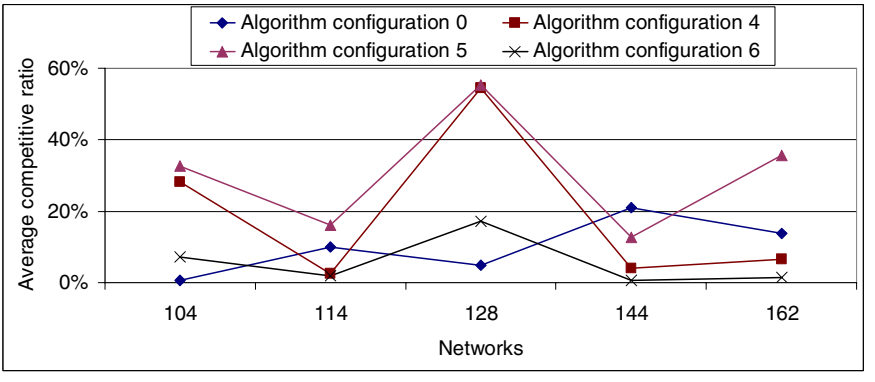


Fig. 3. Average competitive ratio of various algorithm configurations as a function of network. Configuration 0 denotes the LRH algorithm, configurations 4-6 denote versions of EA and HEFAN without init solution.

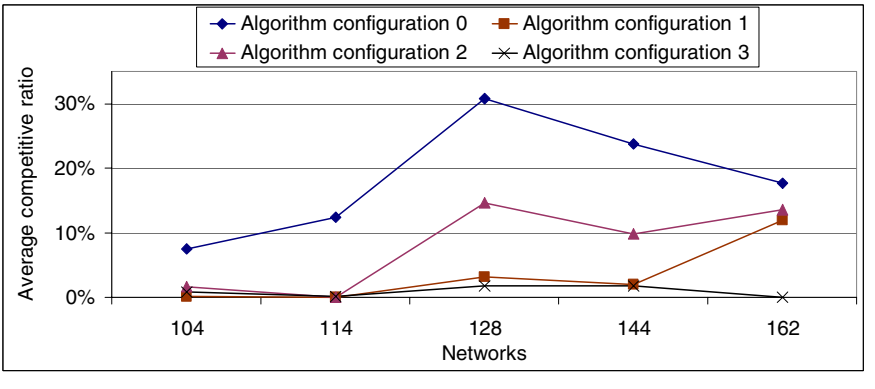


Fig. 4. Average competitive ratio of various algorithm configurations as a function of network. Configuration 0 denotes the LRH algorithm, configurations 1-3 denote versions of EA and HEFAN with init solution.

In Fig. 5 we show the number of cases in which a particular algorithm configuration provides the best found solution for experiment one. Fig. 6. presents corresponding results for experiment two.

Summarizing all results presented in Figs. 3-6 the best performance was obtained for configurations which applies HEFAN 2.2. In experiment one, HEFAN 2.2 without init solution proved to be better than LRH algorithm (about 4,5%). It is very significant observation, since – as shown in Fig. 3 – all previous versions of HEFAN and standard EA without starting feasible solution yield much worse results comparing to LRH. Consequently, results show that adding the low level fitness function $L(k)$ (16) seems to be a step in a good direction. Moreover, we can conclude that HEFAN 2.2 is more capable of finding new good fitting route propositions and is less dependent on initialization data than standard EA that is not capable of finding any route propositions. In the case of HEFAN 2.2 with starting solution, the improvement over LRH results was significant (about 17.8%). However, differences between tested algorithms depend on the network topology and demand pattern. As reported on Fig. 5-6 even HEFAN 1.0 and standard EA in some cases can yield the best solution.

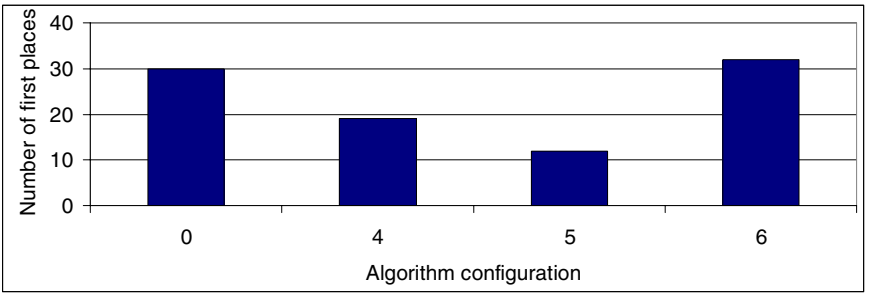


Fig. 5. The number of first places for each algorithm configuration. Configuration 0 denotes the LRH algorithm, configurations 4-6 denote versions of EA and HEFAN without init solution.

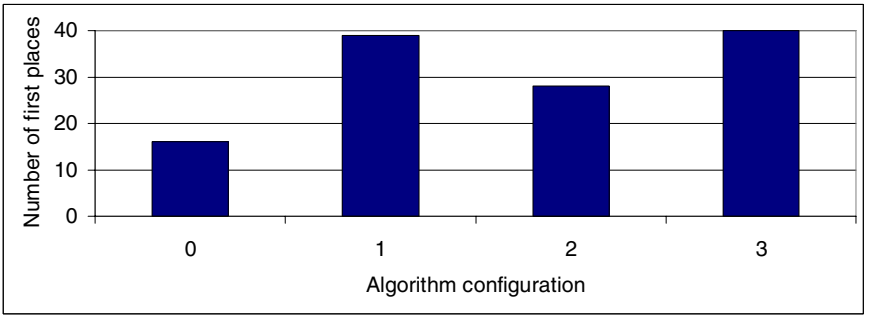


Fig. 6. The number of first places for each algorithm configuration. Configuration 0 denotes the LRH algorithm, configurations 1-3 denote versions of EA and HEFAN with init solution.

Surprisingly the old HEFAN 1.0 results were worse even than standard EA which was not expected. This fact proves that HEFAN 1.0 is not capable of successfully acting in rather large solution space (HEFAN 1.0 was a very effective tool for smaller 10, 14 and 18 - node networks) because it spends too much time for searching new

route proposals instead of trying to find a good route sets made from the available ones. Fortunately HEFAN 2.2 which is searching for new routes in more "aware" way (by using low level fitness function) does not suffer this problem.

5 Conclusion and Further Work

In this work we have proposed new version of evolutionary algorithm for the problem of working routes assignment in survivable MPLS networks. Results of simulation tests have shown that the HEFAN version 2.2 (using the low level fitness function $L(k)$) proposed in this paper, outperforms HEFAN version 1.0 [8-9] and the LRH [3] algorithm. However, there is still space for further improvement, especially in high and low level organization. The low level should be able to remove some of it's individuals (routes) from routes database – it is possible that there is a moment when a large number of unused routes in the routes database slows the algorithm down forcing it to compute the low level fitness function value for routes that are not good. The high level of the algorithm faces a different problem – a very large number of genes necessary to code the whole solution – what causes the EA effectiveness drops down. This is also one of possible reasons why the HEFAN algorithm effectiveness is significantly lower when it is not initiated with a feasible solution. Some possible solution to this problem may be the messy–GA idea, where individuals does not have to contain all necessary genes. The other problem is the high population concentration – in the presented experiments only a small part of all possible solutions are feasible. This means that another directions of the algorithm improvement is adding some "prevent population concentration" mechanisms like parallel evolution.

References

1. Elbaum, R., Sidi, M.: Topological Design of Local Area Networks Using Genetic Algorithms. IEEE/ATM Transactions On Networking 5, 766–778 (1996)
2. Ericsson, M., Rescende, M., Pardalos, P.: A Genetic Algorithm for the weight setting problem in OSPF Routing. J. of Combinatorial Optimization 6, 299–333 (2002)
3. Gavish, B., Huntler, S.: An Algorithm for Optimal Route Selection in SNA Networks. IEEE Trans. Commun. COM-31(10), 1154–1160 (1983)
4. Grover, W.: Mesh-based Survivable Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking. Prentice Hall, New Jersey (2004)
5. Pióro, M., Medhi, D.: Routing, Flow, and Capacity Design in Communication and Computer Networks. Morgan Kaufman, San Francisco (2004)
6. Riedl, A.: A Versatile Genetic Algorithm for Network Planning. In: Proceedings of EUNICE'98, pp. 97–103 (1998)
7. Riedl, A.: A Hybrid Genetic Algorithm for Routing Optimization in IP Networks Utilizing Bandwidth and Delay Metrics. In: Proceedings of IPOM (2002)
8. Przewoźniczek, M., Walkowiak, K.: Evolutionary Algorithm for Congestion Problem in Connection-Oriented Networks. In: Gervasi, O., Gavrilova, M., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3483, pp. 802–811. Springer, Heidelberg (2005)

9. Przewoźniczek, M., Walkowiak, K.: Quasi-hierarchical Evolution Algorithm for Flow Assignment in Survivable Connection-Oriented Networks. International Journal of Applied Mathematics and Computer Science 4, 101–116 (2006)
10. Sharma, V., Hellstrand, F(ed.): Framework for MPLS-based Recovery. RFC 3469 (2003)
11. Walkowiak, K.: A New Method of Primary Routes Selection for Local Restoration. In: Mitrou, N.M., Kontovasilis, K., Rouskas, G.N., Iliadis, I., Merakos, L. (eds.) NETWORKING 2004. LNCS, vol. 3042, pp. 1024–1035. Springer, Heidelberg (2004)

An Exact Algorithm for the Minimal Cost Gateways Location, Capacity and Flow Assignment Problem in Two-Level Hierarchical Wide Area Networks

Przemyslaw Ryba and Andrzej Kasprzak

Wroclaw University of Technology, Chair of Systems and Computer Networks
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
{przemyslaw.ryba, andrzej.kasprzak}@pwr.wroc.pl

Abstract. In the paper the problem of designing two-level hierarchical structure of wide area network (WAN) is considered. The goal is to select gateways location, channel capacities and flow routes in order to minimize total cost of leasing capacities of channels of 2nd level network, subject to constraint on maximal acceptable value of average packet delay in hierarchical network. An exact algorithm is proposed. Some results of computational experiments are reported. Based on computational experiments, several properties of the considered problem, important from practical point of view, are formulated.

Keywords: WAN, Hierarchical Network, Algorithm, Branch and Bound.

1 Introduction

Design of the wide area networks (WAN) containing hundreds of hosts and communication links is challenging task. Customer's requirements include requirements for availability, scalability, affordability, security, and manageability. Many customers also want to specify a required level of network performance. To meet these needs, difficult network design choices and tradeoffs must be made. Algorithms which are suitable for small and moderate-sized networks, when applied directly to large networks, become costly (from computational point of view) and sometimes infeasible. Design of huge WANs should be split functionally to make the project more manageable. Using a hierarchical design with its modular structure can facilitate not only network design, but also future changes and identification of failures. In hierarchical WAN nodes are grouped in 1st level clusters, which in turn are grouped into 2nd level cluster and so on. The communication network of each cluster can be designed separately. In each 1st level cluster special communication nodes called "gateways" are selected. Gateways provide communication between nodes from different 1st level networks. Traffic between nodes in the same cluster uses paths restricted to local communication network. Traffic between nodes in different 1st level networks is first sent to local gateway, then via 2nd level network of gateways is sent to gateway located in destination 1st level network to finally reach the destination node [1]. Example of structure of the two-level hierarchical wide area network is presented in the Fig. 1.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 343 – 352, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In this paper problem of designing the two-level hierarchical wide area network in order to minimize total cost of leasing capacities of channels of 2nd level network subject to delay constraint is considered. The Gateways Location, Capacity and Flow Assignment problem with cost criterion is formulated as follows:

given: topology of 1st level networks and 2nd level network, sets of potential gateways locations, set of potential 2nd level network channels capacities and costs (i.e. discrete cost-capacity function), traffic requirements,

minimize: leasing cost of channel capacities of 2nd level network,

over: gateways locations, 2nd level network channel capacities, multicommodity flow (i.e. routing),

subject to: multicommodity flow constraints, channel capacity constraints, delay constraint in hierarchical network.

Discrete cost-capacity function considered here is justified for the reason that channels capacities can be chosen from the sequence defined by ITU-T (International Telecommunication Union – Telecommunication Standardization Section) recommendations. Such formulated problem is NP-complete. Proof of NP-completeness of considered problem can be found in section 3 of this paper.

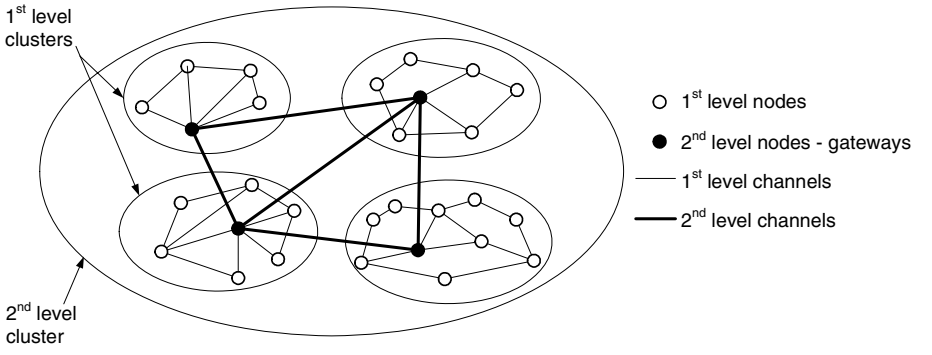


Fig. 1. Structure of two-level hierarchical wide area network

Some algorithms for hierarchical network design can be found in literature [2, 3, 4]. In the paper [3] problem limited to tree topology of hierarchical network is presented. Algorithm for router location which includes simplified topology assignment without allocating capacities to channels, is presented in the paper [2]. Also, in the paper [4] presenting algorithm for interconnecting two WANs – assigning capacities to channels is not considered. In our earlier papers we presented heuristic algorithm for solving problem stated above [5] and exact algorithm for solving similar problem [6] minimizing average packet delay in hierarchical network subject to budget constraint. Solution of the problem presented in [6] can help to design network with best possible performance given budget. In this paper we propose algorithm which may be used to design minimal cost hierarchical network meeting specified performance requirements (constraint on maximal average packet delay).

Heuristic algorithm presented in the paper [5] may be used whenever an optimal solution is not a must. However, it is known by experience that results obtained by heuristic algorithms are worse from optimal results by several percent. Since design of wide area networks is very expensive, each improvement of obtained results can lead to significant savings in costs of network maintenance, e.g. costs of leasing lines from networking services provider [7]. Such improvement can be obtained by using the exact algorithm. Then, constructing exact algorithms is very important from economical point of view. Moreover, optimal solutions obtained by the exact algorithm may be used to evaluation of results obtained using heuristic algorithms.

2 Problem Formulation

Consider a network with *n* nodes grouped in *K* clusters. The communication network of each cluster (1st level networks) is denoted by S_1^l, $l = 1,..., K$. Let N_1^l be set of nodes and L_1^l set of channels in the network S_1^l. In order to set up 2nd level network S_2 connecting 1st level networks, in each network S_1^l given number of gateways is to be located. Let y_a^g be the discrete variable for choosing one of available locations for gateway g defined as follows: $y_a^g = 1$, if gateway g is located in node a and $y_a^g = 0$, otherwise. Each gateway must be placed in exactly one node, thus it is required to satisfy following condition:

$$\sum_{a \in J_g} y_a^g = 1, \quad g \in H^l, \quad l = 1,..., K \tag{1}$$

where H^l is set of gateways to be placed in network S_1^l and J_g is set of possible locations for gateway g.

Let m be number of channels between gateways and let p be total number of channels. For each channel between gateways, capacity must be chosen from the set of available values $C^i = \{c_1^i, ..., c_{s(i)}^i\}$, $i = 1,..., m.$ Values of capacities in each set C^i are ordered in the following way: $c_1^i > c_2^i > ... > c_{s(i)}^i$. Let x_k^i be the discrete variable for choosing one of available capacities for channel i defined as follows: $x_k^i = 1$, if the capacity c_k^i is assigned to channel i and $x_k^i = 0$ otherwise. Since exactly one capacity value from the set C^i must be chosen for channel i, the following condition must be satisfied:

$$\sum_{k=1}^{s(i)} x_k^i = 1 \quad \text{for} \quad i = 1, ..., m \tag{2}$$

Let d_k^i be the cost of leasing capacity c_k^i for channel i defined as follows:

$$d_k^i = \sum_{a \in J_{g1}} \sum_{b \in J_{g2}} y_a^{g1} y_b^{g2} d_{ab}^{ik} \tag{3}$$

where g_1 and g_2 are gateways adjacent to channel i, d_{ab}^{ik} is cost of leasing capacity c_k^i of channel i between gateway g_1 located in node a and gateway g_2 located in node b.

Let $X_r^{'}$ be the permutation of values of variables x_k^i, $i = 1,...,m$ for which the condition (2) is satisfied, and let X_r be the set of variables which are equal to 1 in $X_r^{'}$. Similarly, let $Y_r^{'}$ be the permutation of values of all variables y_a^g for which condition (1) is satisfied and let Y_r be the set of variables which are equal to 1 in $Y_r^{'}$. The pair of sets (X_r, Y_r) is called a selection. Each selection (X_r, Y_r) determines locations of gateways and channels capacities in the 2nd level of hierarchical network. Let $\Re$ be the family of all selections.

Let $d(X_r, Y_r)$ be the total cost of leasing capacities of channels of 2nd level network, in which values of channels capacities are given by X_r and locations of gateways are given by Y_r.

$$d(X_r, Y_r) = \sum_{x_k^i \in X_r} x_k^i d_k^i \tag{4}$$

Then, the considered gateway location, flow and capacity assignment problem in the two-level hierarchical wide area network can be formulated as follows:

$$\min_{(X_r, Y_r)} d(X_r, Y_r) \tag{5}$$

subject to:

$$(X_r, Y_r) \in \Re \tag{6}$$

$$T(X_r, Y_r) \le T^{\max} \tag{7}$$

where $T(X_r, Y_r)$ denotes total average delay per packet in two-level hierarchical network, given by Kleinrock's formula [8] and $T^{\max}$ its maximal acceptable value of that delay.

3 The Branch and Bound Algorithm

Assuming that $|H^l| = 1$ for $l = 1,...,K$, the problem (5-7) becomes the "capacity and flow assignment problem" which is known as a NP-complete [9]. Since the problem (5-7) is more general, it is also NP-complete. Then, the branch and bound method can be used to construct the exact algorithm for solving the considered problem. The detailed description of the calculation scheme of the branch and bound method may be found in the papers [10, 11].

Starting with the initial selection (X_1, Y_1) we generate a sequence of selections. The variables from the sets X_1 or Y_1 are called normal. The variables, which do not belong to X_1 or Y_1 are called reverse. A replacement of any normal variable by the reverse variable is called complementing. The generation of new selection involves the choice of the certain normal variable from X_1 or Y_1 for complementing; it is called the branching rules. The choice of the normal and reverse variables is based on local optimization criterion. For each selection (X_r, Y_r) lower bound is calculated to check the possibility of the generation of the selection (X_s, Y_s) with less value of a criterion d than already found. If such (X_s, Y_s) does not exist, (X_r, Y_r) is abandoned and all its possible successors and next backtrack is made to the predecessor from which (X_r, Y_r) was generated. If new selection (X_s, Y_s) is generated from (X_r, Y_r) by complementing the normal variable by the reverse variable, then the reverse variable is constantly fixed. This reverse variable cannot be complemented by any other in every possible successor of (X_s, Y_s). After backtrack from (X_s, Y_s) to (X_r, Y_r) by reverse variable of certain normal variable from (X_r, Y_r) this normal variable is momentarily fixed. So, for each (X_r, Y_r), there is a set F_r of constantly fixed and set F_r^t. of momentarily fixed variables. The reverse variables in F_r are constantly fixed. Each momentarily fixed variable in F_r^t is the reverse variable abandoned during backtracking process. Variables, which do not belong to F_r or F_r^t are called free in (X_r, Y_r). Algorithm terminates, after backtrack from (X_1, Y_1).

3.1 Branching Rules

The purposes of branching rules is to find the normal variable from the selection (X_r, Y_r) for complementing and generating a successor (X_s, Y_s) of the selection (X_r, Y_r) with the least possible value of the criterion function (4) without violating constraint (7). To take into account the constraint (7) we proposed the auxiliary local criterion: $d(X_r, Y_r) + \alpha \cdot T(X_r, Y_r)$ [5], where coefficient α converts value of average packets delay to cost expressed in [€/month]. Choice of variables x_k^i and x_j^i for complementing is made using modified criterion Δ_{kj}^i usually used in classical CFA problem, formulated as follows [7, 11]:

$$\Delta_{kj}^i = \begin{cases} \dfrac{\alpha}{\gamma}\left(\dfrac{f_i}{c_k^i - f_i} - \dfrac{f_i}{c_j^i - f_i} \right) + \left(d_k^i - d_j^i \right) & \text{for } f_i < c^i \\ \infty \quad \text{otherwise} \end{cases} \tag{8}$$

First term of the sum in expression (8) estimates change of average packet delay in the network caused by change of capacity of channel *i*. Second term represents change of the cost of the channel *i* resulting from change of the channel capacity.

Choice of variables y_a^g and y_b^g for complementing is evaluated using criterion δ_{ab}^g:

$$\delta_{ab}^g = \begin{cases} \dfrac{\alpha}{\gamma} \displaystyle\sum_{i \in L_1^l} \left(\frac{\tilde{f}_i}{c^i - \tilde{f}_i} - \frac{f_i}{c^i - f_i} \right) + \sum_{i:\langle g, g_2 \rangle \in V^g} \left(d_{eb}^{ij} - d_{ea}^{ij} \right) & \text{for } \tilde{f}_i \le c^i \\ \infty \quad \text{otherwise} \end{cases} \tag{9}$$

First term of the sum in expression (9) estimates change of average packet delay in the network caused by change of routing in the network. The flow $\tilde{\mathbf{f}}$ is constructed as follows: the flow from all nodes in network S_1^l sent through gateway located in node a is moved from the routes leading to node a to the routes leading to node b. If the conditions $\tilde{f}_i \le c^i$ are satisfied for every channel in the network S_1^l the flow $\tilde{\mathbf{f}}$ is feasible. The second term of the sum estimates change of the leasing cost of channels adjacent to the gateway g of which location changes.

For complementing should be chosen the pair of variables for which the value of criterion δ_{ab}^g or Δ_{kj}^i is minimal.

3.2 Lower Bound

For each selection (X_r,Y_r), lower bound is calculated to check the possibility of the generation of the selection (X_s,Y_s) with lower value of the criterion function (5) than already found. The lower bound LB_r of the criterion function (5) for each possible successor (X_s,Y_s) of the selection (X_r,Y_r), may be obtained by relaxing some constraints in the problem (5-7). To find the lower bound LB_r we assume that cost of each channel i, which capacity is not fixed yet, equals to the cost of the channel with lowest available capacity from the set C^i and lowest possible length of the channel i. Length of the channel i depends on distance between gateways connected by the channel i. so each pair of possible gateway locations is examined. Lower bound LB_r is the sum of such redefined cost of network channels, thus is the lowest value of the network cost, which can be generated from selection (X_r,Y_r).

4 Heuristic Algorithm

The presented exact algorithm involves the initial selection $(X_1,Y_1) \in \Re$ for which the constraints (6) and (7) are satisfied [12]. Moreover, the initial selection should be the near-optimal solution of the problem (5-7). To find the initial selection algorithm presented in [5] is used. Computation scheme of the heuristic algorithm is shortly presented below.

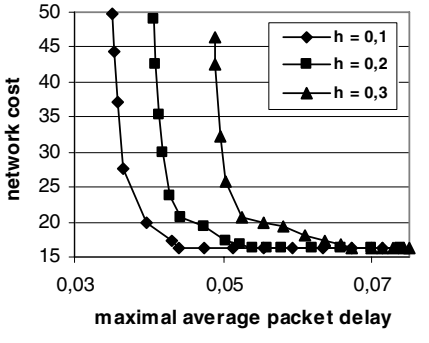
Initially set Y_1 is calculated in such a way that total cost of leasing channels of 2nd level network is minimal. For each channel i, capacity is set to maximal value from the set C^i. If for such constructed network constraint (7) is violated, then one of gateways location is changed. Choice of gateway to change location is performed using expression (9). After each change of gateway location, new value of average delay per packet is compared with one already found. Location with higher value of average

delay per packet is abandoned and removed from set of potential gateway locations. Operation is repeated until constraint (7) is satisfied. After that, capacities of selected channels are decreased to obtain network with lower leasing cost satisfying constraint (7), then current cost of the network $d(X_r, Y_r)$ is compared with already found best solution d^*. If current cost of leasing channels capacities is lower, then d^* is updated. If constraint (7) is violated then operation of changing gateway locations is repeated. Algorithm terminates, when for each gateway, the set of possible locations is empty or there is no channel which capacity can be decreased.

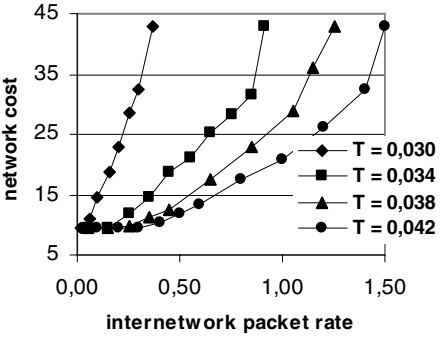
This heuristic algorithm may be also used to design of the hierarchical WAN when the optimal solution is not necessary.

5 Computational Results

The presented algorithm was implemented in C++. Extensive numerical experiments have been performed with this algorithm for number of different network topologies and for many possible gateways number and locations. The experiments were conducted with two main purposes in mind: first, to examine the impact of various parameters on solutions to find properties of the problem (5-7) important from practical point of view and second: to test the computational efficiency of our algorithm. We examined the impact of two parameters: maximal acceptable value of average packet delay T^{max} in the two-level hierarchical network and internetwork packet rate h on the network cost d.



(a) Influence of value T^{max}



(b) Influence of internetwork flow rate h

Fig. 2. Impact of selected parameters on the cost of the network

The Fig. 2(a) shows the typical dependence of the network cost on value $T^{\max}$. We can observe that higher values of $T^{\max}$ result in lower network cost. the dependence of leasing cost of channel capacities d on $T^{\max}$ is decreasing function and that there exists such value $\hat{T}^{\max}$, that the problem (5-7) has the same solution for each $T^{\max}$ greater or equal to $\hat{T}^{\max}$.

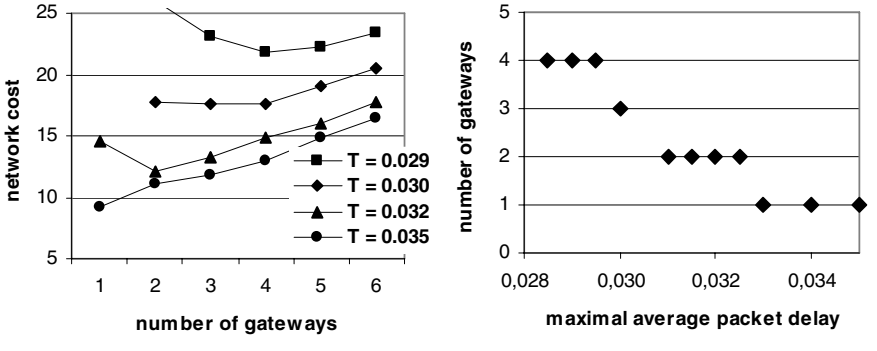
Moreover, it follows from results obtained for many considered networks, that the cost of the network obtained by solving the problem (5-7) in term of maximum acceptable average packet delay $T^{\max}$ is the function of the following class:

$$d = \frac{u_1}{T^{\max} + u_2} + u_3 \tag{10}$$

where u_1, u_2 and u_3 are constant coefficients. To find the function (10) for some hierarchical network it is enough to know only the values of u_1, u_2 and u_3, which can be computed by solving the identification problem, e.g. using the Least Squares Method.

The Fig. 2(b) shows the impact of value of internetwork packet rate on the cost of the network obtained by solving the problem (5-7). We can see that higher packet rate with constant value of quality requirement (maximal acceptable value of average packet delay in the network) induces higher network cost. The network cost increases faster for lower values of $T^{\max}$, i.e. higher quality demands.

In the Fig. 3 we presented results of experiments examining impact of number of gateways located in the 1st level network on the total cost of the hierarchical network meeting specified performance requirements. In the Fig. 3(a) we can observe that effect obtained by adding another gateway to the 1st level network depends on number of gateways already located in the network and also on the value of maximal acceptable average packet delay in the hierarchical network. Optimal number of gateways, i.e. number of gateways for which total cost of the network is minimal, depends on the maximal acceptable value of average packet delay $T^{\max}$.



(a) Network cost versus number of gateways

(b) Optimal number of gateways versus T^{max}

Fig. 3. Results of experiments with number of gateways in the 1st level network

Relationship between optimal number of gateways and value $T^{\max}$ is presented in the Fig. 3(b). We can observe that increasing value of maximal acceptable average packed delay in the network makes the optimal number of gateways in the 1st level network smaller.

In order to evaluate the computational efficiency of the presented algorithm we performed experiments for a large number of networks. For each network we recorded the number of iteration of the algorithm and then compared them for all

considered networks. To compare the results obtained for different hierarchical wide area networks topologies we introduce the normalized maximal acceptable average packet delay:

$$\overline{T}^{\max} = \frac{T^{\max} - T_L^{\max}}{T_H^{\max} - T_L^{\max}} \cdot 100\% ,$$

where $T_L^{\max}$ is the minimal value of $T^{\max}$, such that the problem (5-7) has no solution for $T^{\max} < T_L^{\max}$ and $T_H^{\max}$ is such value of $T^{\max}$ that for $T^{\max} > T_H^{\max}$ solutions of problem (5-7) are equal to solutions of problem (5-6).

Moreover, let $\Phi^i\left(\overline{T}^{\max}\right)$ be the number of iterations of the branch and bound algorithm to obtain the optimal value of network cost *d* for normalized budget equal to $\overline{T}^{\max}$ for *i*-th considered network topology. Let

$$\Phi(u,v) = \frac{1}{Z}\sum_{i=1}^{Z}\left(\sum_{\overline{T}^{\max}\in[u,v]}\Phi^i(\overline{T}^{\max}) \Big/ \sum_{\overline{T}^{\max}\in[0,100]}\Phi^i(\overline{T}^{\max})\right)\cdot 100\%$$

be the arithmetic mean of the relative number of iterations for $\overline{T}^{\max} \in [u,v]$ calculated for all considered network topologies, where Z is the number of considered networks. Fig. 3 shows the dependency of Φ on ranges [0%, 10%), [10%, 20%), ..., [90%, 100%] of normalized value $\overline{T}^{\max}$. It follows from Fig. 3 that the presented algorithm is most effective from computational point of view for $\overline{T}^{\max} \in [30\%, 100\%]$.

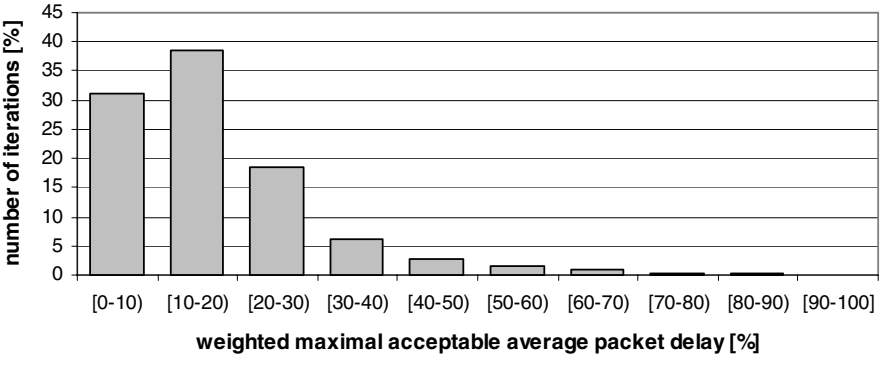


Fig. 4. Performance of presented algorithm for distinct ranges of value $\overline{T}^{\max}$

6 Conclusions

In the paper the exact algorithm for solving the gateway location and network topology assignment problem in two-level hierarchical network to minimize total cost of leasing channel capacities is presented. The high costs of WANs design justify use of computationally costly exact algorithms. Improvement of results obtained by heuristic

algorithms, can lead to significant savings in costs of network maintenance. That makes our algorithm important from economical point of view. Moreover, optimal solutions obtained using our algorithm can be used to evaluate results obtained using various heuristic algorithms. From computational experiments presented in the paper (Fig. 3) follows conclusion that the optimal number of gateways located in 1st level network depends on requirements on network performance. To satisfy high demands on network performance higher number of gateways is required.

This work was supported by a research project of The Polish State Committee for Scientific Research in 2005-2007.

References

1. Kleinrock, L., Kamoun, F.: Optimal Clustering Structures for Hierarchical Topological Network Design of Large Computer Networks. Networks 10, 221–248 (1980)
2. Liu, Z., Gu, Y., Medhi, D.: On Optimal Location of Switches/Routers and Interconnection. Technical Report, University of Missouri-Kansas City (1998)
3. Saha, D., Mukherjee, A.: On the multidensity gateway location problem for multilevel high speed network. Computer Communications 20, 576–585 (1997)
4. Liang, S.C., Yee, J.R.: Locating Internet Gateways to Minimize Nonlinear Congestion Costs. IEEE Transactions On Communications 42, 2740–2750 (1994)
5. Ryba, P., Kasprzak, A.: An Approximate Algorithm for the Minimal Cost Gateways Location, Capacity and Flow Assignment in Two-Level Hierarchical Wide Area Networks. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J.J. (eds.) ICCS 2006. LNCS, vol. 3991, pp. 100–107. Springer, Heidelberg (2006)
6. Ryba, P., Kasprzak, A.: The Gateways Location and Topology Assignment Problem in Hierarchical Wide Area Networks: Algorithms and Computational Results, In: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganà, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3982, pp. 1100–1109. Springer, Heidelberg (2006)
7. Kasprzak, A.: Topological Design of the Wide Area Networks. Wroclaw University of Technology Press, Wroclaw (2001)
8. Fratta, L., Gerla, M., Kleinrock, L.: The Flow Deviation Method: an Approach to Store-and-Forward Communication Network Design. Networks 3, 97–133 (1973)
9. Markowski, M., Kasprzak, A.: An exact algorithm for host allocation, capacity and flow assignment problem in WAN. In: WITASI 2002, pp. 73–82. Kluwer Academic Publ, Dordrecht (2002)
10. Walkowiak, K.: A New Method of Primary Routes Selection for Local Restoration. In: Mitrou, N.M., Kontovasilis, K., Rouskas, G.N., Iliadis, I., Merakos, L. (eds.) NETWORKING 2004. LNCS, vol. 3042, pp. 1024–1035. Springer, Heidelberg (2004)
11. Markowski, M., Kasprzak, A.: The Web Replica Allocation and Topology Assignment Problem in Wide Area Networks: Algorithms and Computational Results. In: Gervasi, O., Gavrilova, M., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3483, pp. 772–781. Springer, Heidelberg (2005)
12. Wolsey, L.A.: Integer Programming. Wiley-Interscience, New York (1998)

Implementing and Optimizing a Data-Intensive Hydrodynamics Application on the Stream Processor

Ying Zhang, Gen Li, and Xuejun Yang

National Laboratory for Paralleling and Distributed Processing, School of Computer
National University of Defense Technology, Changsha, Hunan, 410073, P.R. of China
zhangying@nudt.edu.cn

Abstract. Several representative scientific computing applications have been mapped on the stream processor. But most of them are computation-intensive kernels or synthetic benchmarks. In this paper, we implement and optimize a complete data-intensive hydrodynamics application, QNJ-5, on the stream processor which is designed for computation-intensive applications. Different from other stream programs, how to relieve memory access pressure is especially important to this stream program. Simulation results show that StreamQNJ-5 gets an ultimate speedup of 2.97 and 1.11 over original FORTRAN QNJ-5 on a Xeon and Iantium processor, respectively.

Keywords: data-intensive scientific computing, stream processor, kernel join, loop-carried stream reusing, stream transpose.

1 Introduction

Of all emerging architectures, the stream processor has absorbed a lot of scientific computing researchers' attention[1-8], due to its capacity of high parallel computing, memory system, low power and low cost[9-11]. The stream processor is absolutely different from other conventional processors whether from programming model or from the architecture.

Stream programming model organizes applications as two levels: kernel-level and stream-level[12-15]. A kernel is a computation-intensive function that operates on sequences of records called streams. Each kernel takes streams of records as input and produces streams as output. Kernels are written using a C-like language called KernelC. The stream-level program declares the streams and defines the high-level control- and data-flow between kernels. Stream-level programs are written using a programming language extension called StreamC intermixed with C++. Stream programming model exposes three level parallelisms of the stream architecture to programmers: Instruction-Level Parallelism (ILP), Data-Level Parallelism (DLP) and Task-Level Parallelism (TLP).

Usually, the stream processor (as shown in fig. 1) acts as a computation-intensive application accelerator of a host. Both its ALU organization and the memory system are different from those of conventional architecture. There are N clusters controlled

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 353 – 366, 2007.
© Springer-Verlag Berlin Heidelberg 2007

by a single Micro Controller to work in SIMD mode, and every cluster consists of M ALUs (in the stream processor prototype Imagine[16][17] developed by Stanford University, N = 8 and M = 6). At the same time, the stream processor has three level memory hierarchies – local register files (LRF) near ALUs, exploiting locality in kernels; global stream register files (SRF), exploiting producer-consumer locality between kernels; and streaming memory system, exploiting global locality.

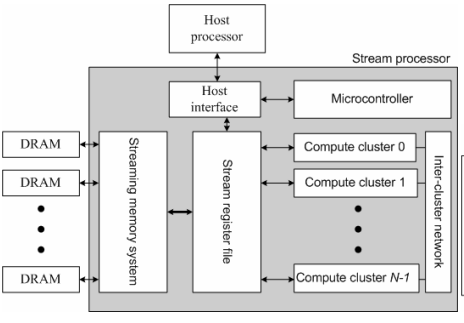


Fig. 1. Block diagram of a stream processor

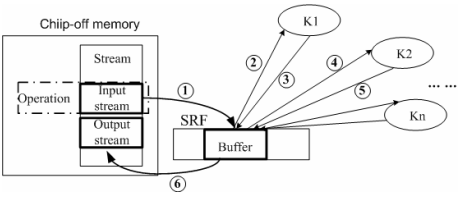


Fig. 2. Stream flowing across the memory system

Fig. 2 shows a stream flows across three level memory hierarchies during the execution of a stream program. First, the stream is loaded from off-chip memory into the SRF and distributed among buffers in SRF. Then some records of it are loaded from SRF to LRF as operands of a kernel, and then following records are processed. Repeat doing the same kernel until the whole stream are consumed. If about this stream there is producer-consumer locality between this kernel and its later kernels, the stream is reserved in the SRF. Otherwise, they are stored back to off-chip memory.

Several representative scientific computing applications have been developed on the stream processor, such as QRD, FEM and streamFLO[4][5][6]. But most of them are computation-intensive kernels or synthetic benchmarks. To evaluate the performance of data-intensive applications on the stream processor which is aimed at computation-intensive applications originally, we map and optimize a complete hydrodynamics application, QNJ-5, which requires *O(n)* floating-point operations to solve *O(n)* data.

There are two reasons for our mapping a data-intensive application on the stream processor. For one thing, there are a great lot of data-intensive scientific applications, such as BLAS-1 and BLAS-2[20]. For the other thing, while computation-intensive kernels are chosen to run on the stream processor, usually acting as a coprocessor of a host, what should be done with data-intensive kernels among computation-intensive kernels? Generally, it is better to run successive kernels on the stream processor, due to the expensive communications between the host and the stream processor. Additionally, it is very important to optimize a computation-intensive application on the

stream processor for better performance. So it is necessary to evaluate and optimize the data-intensive scientific application on the stream processor.

This paper presents our implementation and optimization of QNJ-5 on the stream processor. Firstly, our mapping methods are given. Then, the generated stream program, called StreamQNJ-5, is performed on Isim[16][17], a cycle-accurate stream simulator. Three optimizations and corresponding performance evaluation are given at last, special for the stream processor and the stream programming model. Compared with original FORTRAN program on a Xeon and Itanium processors, StreamQNJ-5 ultimately achieves an ultimate speedup of 2.97 and 1.11 respectively.

The remainder of this paper describes and evaluates our implementation and optimization of StreamQNJ-5. In Section 2 we describe our implementation of StreamQNJ-5 on the stream processor. In section 3 we evaluate and analyze the performance of StreamQNJ-5 on Isim. In section 4 we optimize our implementation special for the stream processor, and evaluate and analyze our optimization. Section 5 discusses what must be done before the stream processor can support data-intensive real scientific computing applications well and section 6 makes a conclusion.

2 QNJ-5

QNJ-5 application is a nonlinear algebra solver of two-dimensional nonlinear diffusion of hydrodynamics, and the nonlinear diffusion equation is:

$$QU_t = B_1U_{xx} + B_2U_{yy} + B_3U_x + B_4U_y + eU + g \ .$$

The disperse scheme of QNJ-5 application is 5-point fully-hidden. The application uses the solver for the systems of nonlinear algebraic equations, which is made up of neutron local linearization method and the solver for sparse systems of linear equations. The solver for sparse systems of linear equations is QMRCGSTAB algorithm with diagonal preconditioners. QMRCGSTAB utilizes subspace iteration method to solve large nonsymmetric linear systems[18]. In our mapping, a 800 800 matrix is solved by iterating 2500 times.

QNJ-5 has intensive memory access and there are only *O(n)* floating-point operations on *O(n)* data loaded from off-chip memory; QNJ-5 has complex data access traces, such as only left top matrix or several medium diagonal lines being processed; QNJ-5 has strong data dependency, and for example, several matrixes is multiplied by a dot product of two former long vectors. All these make it hard to implement high performance StreamQNJ-5.

Additionally, different from other stream programs, for mapping QNJ-5, it is especially important to reduce memory accesses. Then we'll take example for code in fig. 3(a) to describe our implementation, which is also fit for mapping other scientific stream programs, especially for data-intensive applications, to the stream processor.

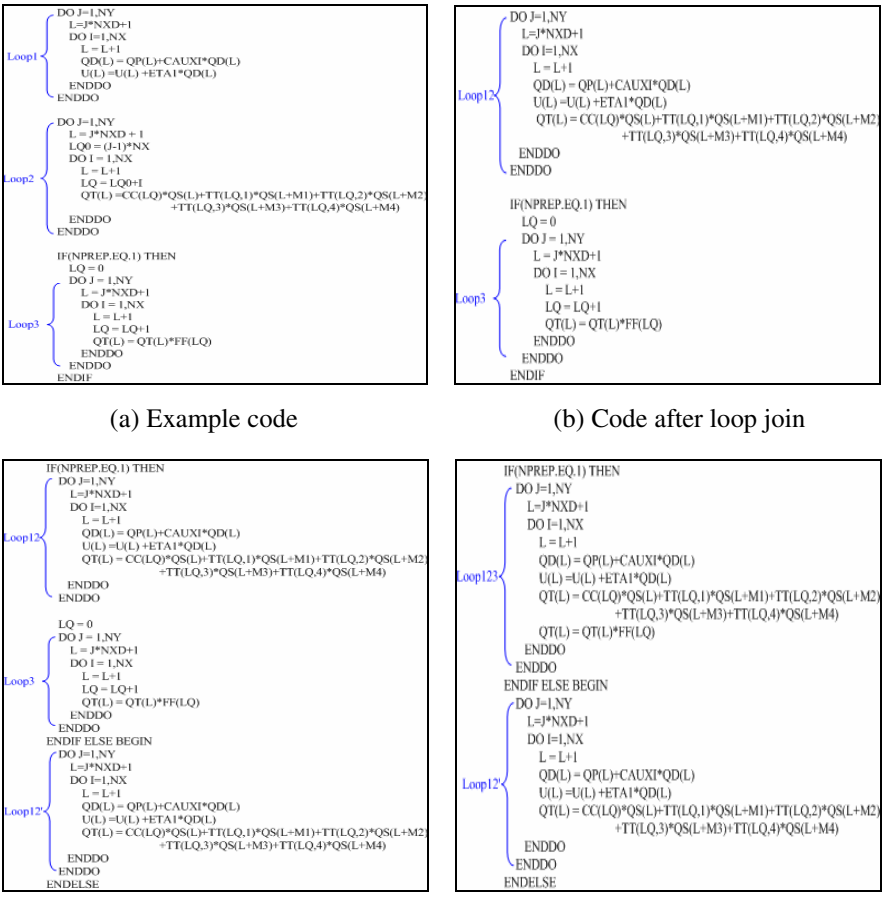


Fig. 3. Example code

2.1 Stream-Level

The stream-level program declares the streams and defines the high-level control- and data-flow between kernels. When implementing a stream-level program, we should know the data access methods and data dependencies of the selected code first. Then, aiming at improving calculations/memory access ratio and cluster parallelism, we organize streams by way of data dependence analysis and loop transformations. Now, we give the transformations used for code in fig. 3(a).

- Loop fusion. The method of loop fusion involves joining free adjacent loops which have the same number of loops and loop bounds. This method can enhance the temporal locality and operation density of joined loops. As for the stream processor architecture, this method exploits more LRF locality and cluster ILP. Loop1 and

Loop2 in fig. 3(a) can be joined together because they are free loops and have the same number of loops and loop bounds, while Loop2 and Loop3 can't because Loop3 isn't a free loop. The code after loop join transformation is shown in fig. 3(b). Generally, if two loops are of different loop types but have common operands, we should join them as possible as we can by way of loop peeling, loop normalization, loop spreading, and so on first. A simple example is shown in fig. 4. To make the loops in fig. 4 be transformed by the method of loop fusion, we use loop spreading transformation first.

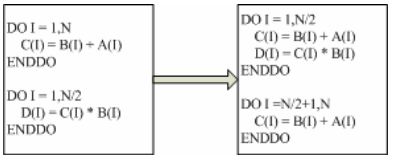


Fig. 4. Loop fusion of different types

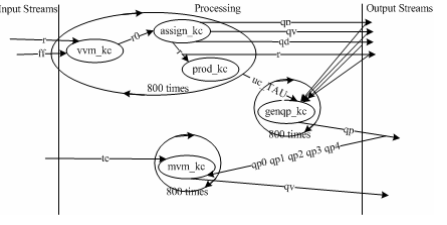


Fig. 5. Part of StreamQNJ-5's kernel flow graph

- Forward condition. If the body of a conditional statement contains loop statements which can be fused with loops near the conditional statement, such as Loop3 and Loop12 in fig. 3(b), we should forward the condition to make loops that can be joined together in a branch body, as shown in fig. 3(c).
- Iteration of loop fusion and forward condition. We use the two methods above repeatedly to make the computations in a loop as many as possible, as shown in fig. 3(d).

2.2 Kernel Level

A kernel is a computation-intensive function that operates on sequences of records called streams. Generally, intensive computations on streams determined above should be organized as kernels. In the example code, computations in Loop123 and loop12' body should be written into kernels respectively.

Fig. 5 shows a part of StreamQNJ-5's kernel flow graph, which is representative of the whole program's kernel flow graph. As shown in the figure, a kernel usually acts as a part of a loop body. That means when running, producer-consumer locality only exists between adjacent kernels in the same loop body or between kernels of successive loop iterations.

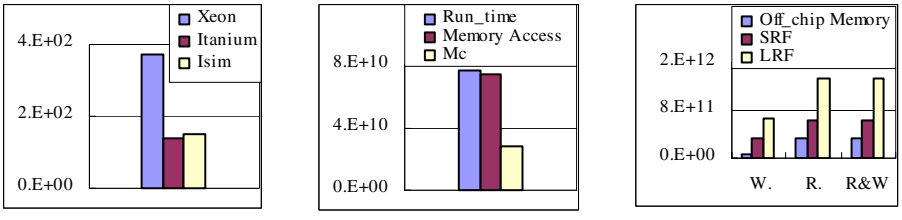
3 Experimental Setup and Performance Evaluation

In our experiments, we use Isim[16][17], a cycle-accurate stream processor simulator supplied by Stanford University, to get the performance of StreamQNJ-5. The baseline

configuration of the simulated stream processor and memory system is detailed in table 1, and is used for all experiments unless noted otherwise. For comparison, FORTRAN QNJ-5 is compiled by Intel's IA64 compiler (with optimization option -O3) and by Intel's IA32 compiler (with max speed optimization option) respectively. The compiled versions of FORTRAN QNJ-5 run on a Xeon processor and on an Itanium 2 processor respectively. The operating frequency of the Xeon processor used here is 2.4GHz, and its L1 and L2 Cache capacity is 16KB and 512KB respectively. The operating frequency of the Itanium 2 processor used here is 1.5GHz, and its L1, L2 and L3 Cache capacity is 16KB, 256KB and 4MB respectively. These two types of processors are the most popular machines used to solve scientific computing now.

Table 1. Baseline parameter of Isim

Parameter		Parameter	
Number of clusters	8	Operating frequency	500MHz
Capacity of LRF	9.6KB	Bandwidth of LRF	544GBps
Capacity of SRF	128KB	Bandwidth of SRF	32GBps
Capacity of off-chip DRAM	4MB	Bandwidth of off-chip DRAM	2.67GBps



(a) Run time of QNJ-5 on Xeon, Itanium and Isim(s)

(b) Active time of main units vs. StreamQNJ-5's run time(Cycles)

(c) Traffics of off-chip memory, SRF and LRF

Fig. 6. Performance evaluation of StreamQNJ-5

Fig. 6(a) lists the performance of QNJ-5 on the Xeon, Itanium and Isim. Isim gets much better performance than the Xeon processor, although Isim's operating frequency is much lower than Xeon's. This is because that: plenty of ALUs effectively exploit StreamQNJ-5's parallelism in a SIMD mode; the efficient overlap between memory accesses and computing relieves the overheads brought by memory accesses.

But Isim gets a little worse performance than the Itanium processor which has been validated successfully for scientific computing applications. There are many reasons for Isim's medium performance. Most of them is that there is too little producer-consumer locality in StreamQNJ-5, which means when a kernel runs its input and output streams must be loaded from or saved to off-chip memory.

Fig. 6(b) shows the time cost by memory accesses and the MC (Micro Controller, MC spends time in controlling clusters to work and MC active time equals to cluster working time). The total time they cost is much longer than StreamQNJ-5's run time, which means the stream processor well overlaps memory access with computation.

The time for memory access weighs 97% of StreamQNJ-5's run time, which means memory access is dominant in StreamQNJ-5.

Fig. 6(c) shows traffics of off-chip memory, SRF and LRF. The ratio between them is 1:1.97:3.04. The stream processor exploits the producer-consumer locality in StreamQNJ-5, because of the ratio between off-chip memory and SRF. But the producer-consumer locality in StreamQNJ-5 is very little. The stream processor also exploits LRF locality in kernels. It is very little either.

Experiment results show that for data-intensive stream program, exploiting producer-consumer locality and reducing off-chip memory traffics are very important. Then we will optimize StreamQNJ-5 with such objects.

4 Implementation Optimizations

Imagine provides two methods, unroll and pipeline, to optimize kernels, and two methods, doUnroll and doSoftwarePipeline, to optimize stream-level programs [14][15]. The implementation above has been optimized by these methods. Then we'll depict optimizations used in StreamQNJ-5 specially. These optimizations can also be used to optimize such applications that have the similar characteristics with QNJ-5.

4.1 Kernel Join

Kernel join involves joining several small adjacent kernels into one big kernel. An example is given in fig. 7. In the original program, *vcvpm_kc*, the first *prod_kc* and the second *prod_kc* are processed sequentially. The total number of their streams is 7 and there is a producer-consumer locality among them about stream *r*. After kernel join optimization, there is only one kernel and it has only 3 streams parameters. The producer-consumer locality has changed into locality in LRF of the generated kernel. All operations of original kernels are executed in one kernel, which enhance the ILP of the joined kernel. Since there are 3 stream parameters in the joined kernel, it can still be joined with other kernels if possible.

The optimization of kernel join benefits stream programs from following aspects:

- Enhancing the locality of LRF. Kernel join change original producer-consumer locality between kernels into the locality of LRF in a kernel. As a result, the traffics between the SRF and LRFs are reduced greatly.
- Improving ILP of a kernel. Kernel join enlarges the scale of a kernel, so that ILP of the kernel is also improved and ALUs of the stream processor can be utilized more effectively.
- Taking advantage of the high bandwidth between the SRF and LRFs. Kernel join increase the number of streams that be loaded from the SRF to the LRFs simultaneously. As a result, this method improves the total bandwidth between SRF and LRFs proportionably.

Kernel join optimizes stream programs well, but this method can't be used under following situations that:

- Two successive kernels, the total number of input/output streams of which exceeds the restriction of the stream processor, can't be joined together. The design of the stream processor architecture determines that the number of a kernel's stream parameters equals to at most the number of clusters. For example, it is 8 in Imagine.
- There is loop-carried flow dependence in the kernels to be joined. If such kernels are joined, the stream program will generate wrong results.

Two special cases that can be optimized by kernel-join are given below. The one case is that if a kernel has less stream parameters, two or more successive loop iterations of stream-level program loop can be joined into one. The other case is that if a kernel has less stream parameters and it can't be joined with adjacent kernels or loop iterations, its number of stream parameters is changed double or more by splitting every stream into two or more equal parts and its loop body processes double or more records at a time. Both cases can get similar performance improvement with kernel join and benefit StreamQNJ-5 greatly.

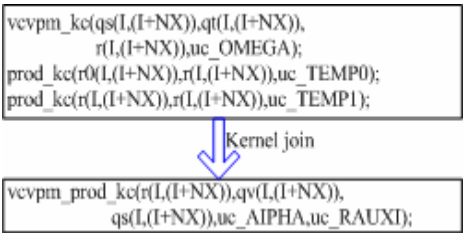
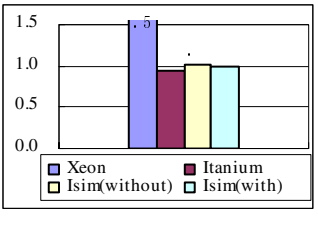
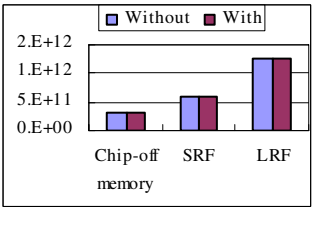


Fig. 7. Example optimized by kernel join



(a) Performance relative to Isim(with) = 1 for each scene



(b) Traffics of off-chip memory, SRF and LRF

Fig. 8. Performance evaluation of kernel join

Fig. 8 shows the performance evaluation of kernel join. Optimized StreamQNJ-5 gets a speedup of 1.02 by kernel join. And it gets a speedup of 2.51 and 0.94 over FORTRAN QNJ-5 on the Xeon and Iantium processor respectively. Because loops in original FORTRAN have been optimized by loop-join, the speedup yielded by this optimization is small. But for a raw stream program, this optimization can get all benefits listed above. Fig. 8(b) shows the total traffics of off-chip memory, SRF and LRF, with and without our optimization. In StreamQNJ-5, the optimization is only

used under the two special cases above, which means there are no different kernels to be joined. So the optimization has no effect on memory hierarchies' traffics.

4.2 Loop-Carried Stream Reusing

Loop-carried stream reusing is defined as that between loop iterations of a stream-level program, input or output streams of kernels in the former iteration can be used as input streams of kernels in the latter iteration. If input streams are reused, we call it input stream reusing. Otherwise, we call it output stream reusing. The essential of stream reusing optimization is to enhance the locality of SRF. Correspondingly, input stream reusing can enhance the producer-producer locality of SRF while output stream reusing can enhance the producer-consumer locality of SRF. Then, we take code in fig. 9 as example to depict our method, where NXD equals to NX+2. In the paper, NX and NY equal to 800.

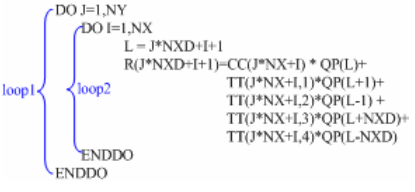


Fig. 9. Example code

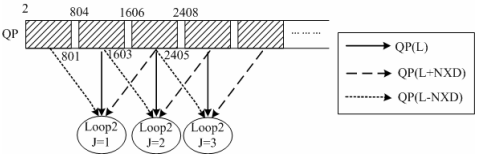
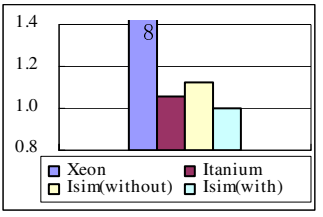


Fig. 10. Data traces of QP(L), QP(L+NXD) and QP(L-NXD)

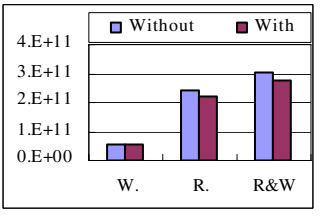
Fig. 10 shows the data trace of QP(L), QP(L+NXD) and QP(L-NXD) participating in loop2 of fig. 9. QP(1606,2405) acts as QP(L+NXD) of loop2 when J=1, as QP(L) when J=2, and as QP(L-NXD) when J=3. So, stream QP can be reused between different iterations of loop1. As a result, if QP(1606,2405) is organized as a stream, it will be in SRF after loop2 with J=1 finishes. When loop2 with J=2 or J=3 runs, it doesn't get stream QP(1606,2405) from off-chip memory but from SRF.

In unoptimized StreamQNJ-5, different array QP references in loop1 are organized as three data streams according their data trace, and operations in loop1 are organized as a kernel. The length of each data stream is 800*800. When running, the stream program must load these three streams into the SRF, the total length of which is 640000*3, nearly three times of QP's length.

After optimized, different array QP references in loop2 are organized as three data streams according their own data trace, operations in loop2 are organized as a kernel, and loop control operations in loop1 are organized as the stream-level program. Thus there would be 800*3 data streams in the stream program loop1, and the length of each is 800. So in loop1, stream QP(L), QP(L+NXD) and QP(L-NXD) of neighboring iterations can be reused. As a result, the stream program only need load 800 data streams with the length of 800 from off-chip memory to SRF, the total length of which is 640000, nearly 1/3 of unoptimized program's.



a) Performance relative to Isim(with) = 1 for each scene



(b) Traffics of off-chip memory, SRF and LRF

Fig. 11. Performance evaluation of loop-carried stream reusing

Fig. 11 shows the performance evaluation of loop-carried stream reusing. Optimized StreamQNJ-5 gets a speedup of 1.13 by loop-carried stream reusing. And it gets a speedup of 2.82 and 1.06 over FORTRAN QNJ-5 on the Xeon and Iantium processor respectively. For QNJ-5, the stream processor gets better performance than the Itanium processor now. This optimization improves the performance greatly.

Fig. 11(b) shows the traffics of off-chip memory with and without loop-carried stream reusing. This method can also cut down off-chip memory traffics proportionably. In StreamQNJ-5, 14994 out of 131467 streams (these streams must be loaded from or saved to off-chip memory) are reused as 4998 streams in SRF. And the off-chip memory traffics become 92.4% of unoptimized StreamQNJ-5. Because StreamQNJ-5 is optimized all by input stream reusing, load memory traffics are cut down obviously while there is no effect on store memory traffics.

There are a lot of applications that can be optimized by loop-carried stream reusing. For a perfect nest of D loops whose body reads elements of the m-dimensional array A twice, if in the Pth level loop the data trace of one array A read references with I_P=i is the same to that of the other array A read references with I_P=i+1 and different to that of the other array A read references with I_P=i, such as the data trace of array QP in loop2 in fig. 9, the two array A read references can be optimized by input stream reusing. Then we give the step of using input stream reusing method to optimize stream organization.

Step A. Organize different array A references in the innermost D-P loops as stream A1 and A2 according their data traces.
Step B. Organize all operations on array A in the innermost D-P loops as a kernel.
Step C. Organize all operations in the outmost P loops as stream-level program.

When the nest with I_P=i of loop P in stream-level program operates on stream A1 and A2, one of them has been loaded into SRF by the former nest, which means that the kernel doesn't get it from off-chip memory but from the SRF. From the feature of the stream processor architecture, we can know the time to access off-chip memory is much larger than that to access the SRF, so the method of stream reusing can improve stream program performance greatly.

The steps to use output stream reusing are analogous to steps above.

4.3 Stream Transpose

Stream transpose involving transposing the distribution of a stream in buffers of SRF in order to make records distributed among successive clusters be distributed on the same cluster.

There are many buffers in SRF, and every buffer is connected with a functional unit to provide data for it, including clusters. Fig. 12 shows the distribution of stream QP of loop2 in fig. 9 among the buffers, assuming that the number of clusters is 8, buffer0 connects with Cluster0, buffer1 connects with Cluster1, and so on. Note that records are interleaved among buffers, which means QP(L), QP(L+1) and QP(L-1) participating in calculations of loop2 are distributed among adjacent clusters.

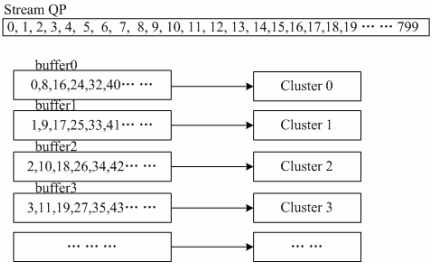


Fig. 12. Distribution of a stream QP among buffers

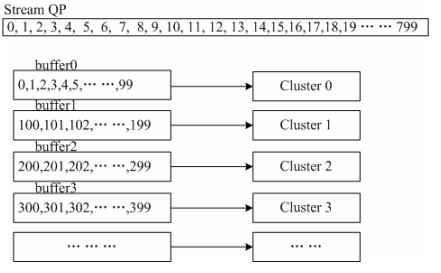


Fig. 13. Distribution of transposed stream QP

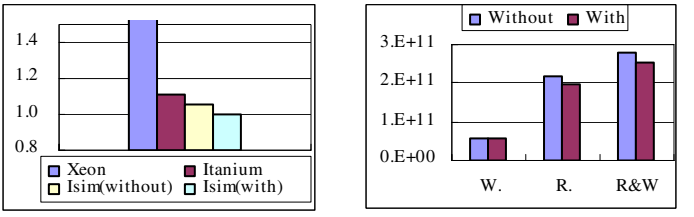
If successive records of the same array participate in the same calculations, such as QP(L), QP(L+1) and QP(L-1) in loop2, such applications can be mapped as follow:

- organizing these records as one stream or organizing different records as different streams according their data trace. If we organize QP(L), QP(L+1) and QP(L-1) as one stream QP'(L), when $cluster_i$ executes operations of loop2, it must communicate with $cluster_{i+1}$ and $cluster_{i-1}$ to gather necessary records on other clusters, which makes inter-cluster communication be the bottleneck of program performance.
- QP(L), QP(L+1) and QP(L-1) are organized as three data stream, then memory access becomes the bottleneck of the stream program.

In StreamQNJ-5 implemented above, we chose the last method. QP(L), QP(L+1) and QP(L-1) are organized as three data stream.

By stream transpose, there is only one stream QP. It consists of array QP(J*NXD, J*NXD+NX+1) in fig. 9. Stream QP is divided into 8 (the number of clusters in a stream processor) parts equally and every part is organized as one record of new stream. That means stream QP consists of only 8 records and all elements of the same record are distribute on the same buffer. As a result, the distribution of stream QP is transposed as fig. 13. QP(L), QP(L+1) and QP(L-1) participating in calculations of loop2 are distributed in the same buffer.

Fig. 14 shows the performance evaluation of stream transpose. Optimized StreamQNJ-5 gets a speedup of 1.05 by stream transpose. And it gets a speedup of



a) Performance relative to Isim(with) = 1 for each scene

(b) Traffics of off-chip memory, SRF and LRF

Fig. 14. Performance evaluation of stream transpose

2.97 and 1.11 over FORTRAN QNJ-5 on the Xeon and Iantium processor respectively. This optimization improves the performance greatly without introducing any communication overheads.

Fig. 14(b) shows the traffics of off-chip memory with and without stream transpose. This method can also cut down off-chip memory traffics proportionably. In StreamQNJ-5, 14994 out of 121471 streams (these streams must be loaded from or saved to off-chip memory) are cut down to 4998 streams. And the off-chip memory traffics become 91.8% of unoptimized StreamQNJ-5.

Because stream transpose change memory access sequences, access conflicts and imbalance in optimized program are much more than in unoptimized program [19]. As a result, the performance improvement is not proportional to the reduction of off-chip memory traffics.

There are a lot of applications that can be optimized by Stream transpose. For a loop, if successive elements of the same array are processed in the loop body, its corresponding stream program can be optimized by stream transpose. Then the general steps of stream transpose are given below:

Step A. If the stream program organizes adjacent records from the same array participating in the same computations as one stream, we can do the next step directly. Otherwise, we must join the streams of the same array first, which means organizing the whole array as a stream.

Step B. Derive a new stream with the streams to be transposed by letting the length of a record equal to $Length_{stream}/N_{cluster}$ and letting the stride equal to $Length_{stream}/N_{cluster}$, where $Length_{stream}$ represents the length of the stream to be transposed, and $N_{cluster}$ represents the number of clusters of the stream processor.

5 Discussion

StreamQNJ-5 gets good performance on the stream processor after finely optimized. For data-intensive applications, if the stream processor wants to get high performance, some key issues should be solved first:

- The compiler should be able to optimize stream programs automatically. For programmers, they can't produce a good stream program until they know the architecture well. So it is important to move programmers' work to the compiler in order to relieve programmers' burden and to generate high performance stream programs. Although

some semi-automatic or automatic optimizations are provided in Imagine, these methods weren't presented specially for the stream programming model and architecture, but inherited from the vector architecture existing years ago.

- The memory system should be improved special for the stream programming model and data-intensive applications. For data-intensive applications, memory accesses cost so much time that all of them can't be overlapped with computation. So it is necessary to improve the memory system, oriented to the quite new stream processor and the stream programming model.

6 Conclusion

In this paper, we implement and optimize a complete data-intensive hydrodynamics application, QNJ-5, on the stream processor which is designed for computation-intensive applications. Our implementation and optimizations for a complete application are given in detail in the paper.

For a data-intensive application, memory accesses take so much time that they can't be overlapped completely. So our implementation and optimizations all aim at to take advantage of the memory hierarchies: Kernel join utilizes the bandwidth between SRF and LRF; Loop-carried stream reusing improves producer-consumer locality in the SRF; Stream transpose reduces memory accesses proportionably by changing three input streams into one without introducing communication overheads.

Measurements of StreamQNJ-5 on Isim show that the architecture is successful in developing such a data-intensive application, because computation kernels in QNJ-5 are popular for data-intensive applications, such as dot product and sparse matrix multiplication. Over the original FORTRAN QNJ-5 on a Xeon and Iantium processor, StreamQNJ-5 gets an ultimate speedup of 2.97 and 1.11 respectively.

Although the stream processor gets good program performance for such applications, there is a great deal work to do to make this architecture and its compiler fit data-intensive applications well as discussed above.

Acknowledgements. We gratefully thank the Stanford Imagine team for the use of their compilers and simulators and their generous help. We also acknowledge the reviewers for their insightful comments. This work is supported by the National High Technology Development 863 Program of China under Grant No. 2004AA1Z2210.

References

1. Dally, W.J., Hanrahan, P., Erez, M., Knight, T.J., Labonté, F., Ahn, J.-H., Jayasena, N., Kapasi, U.J., Das, A., Gummaraju, J., Buck, I.: Merrimac: Supercomputing with Streams, SC, November 2003, Phoenix, Arizona (2003)
2. Erez, M., Ahn, J.H., Jayasena, N., Knight, T.J., Das, A., Labonté, F., Gummaraju, J., Dally, W.J., Hanrahan, P., Rosenblum, M.: Merrimac - Supercomputing with Streams. In: Proceedings of the, SIGGRAPH GP2 Workshop on General Purpose Computing on Graphics Processors, June 2004, Los Angeles, California (2004)

3. Merrimac – Stanford Streaming Supercomputer Project, Stanford University, http://merrimac.stanford.edu/
4. Fatica, M., Jameson, A., Alonso, J.J.: StreamFLO: an Euler Solver for Streaming Architectures, IAA Paper 2004-1090. In: 42nd Aerospace Sciences Meeting and Exhibit Conference, Reno (January 2004)
5. Narayanan, M., Oliker, L., Janin, A., Husbands, P., Li, X.S.: Scientific kernels on VIRAM and imagine media processors. Lawrence Berkeley National Laboratory. Paper LBNL-54908 (October 10, 2002)
6. Erez, M., Ahn, J.H., Garg, A., Dally, W.J., Darve, E.: Analysis and Performance Results of a Molecular Modeling Application on Merrimac, SC'04, Pittsburgh, Pennsylvania, USA (November 2004)
7. Guibin, W., Yuhua, T., et al.: Application and Study of Scientific Computing on Stream Processor. Advances on Computer Architecture (ACA'06), August, Chengdu, China (2006)
8. Jing, D., Xuejun, Y., et al.: Implementation and Evaluation of Scientific Computing Programs on Imagine. Advances on Computer Architecture (ACA'06), Chengdu, China (August 2006)
9. Khailany, B.: The VLSI Implementation and Evaluation of Area- and Energy-Efficient Streaming Media Processors, Ph.D. Thesis, Dept. of Electrical Engineering, Stanford University (2003)
10. Saman Amarasinghe, W.: Stream Architectures. In: Malyshkin, V. (ed.) PaCT 2003. LNCS, vol. 2763, Springer, Heidelberg (2003)
11. Rixner, M.: Stream Processor Architecture. Kluwer Academic Publishers, Boston, MA (2001)
12. Kapasi, U.J., Rixner, S., et al.: Programmable Stream Processor. IEEE Computer, Los Alamitos (2003)
13. Mattson, P.: A Programming System for the Imagine Media Processor. Dept.of Electrical Engineering. Ph.D. thesis, Stanford University (2002)
14. Mattson, P., et al.: Imagine Programming System Developer's Guide (2004)
15. Das, A., Mattson, P., et al.: Imagine Programming System User's Guide 2.0 (June 2004)
16. The Imagine Project, Stanford University, http://cva.stanford.edu/imagine/
17. Kapasi, U.J., Dally, W.J., et al.: The Imagine Stream Processor. In: Processings of the 2002 International Conference on Computer Design (2002)
18. Chan, T.F., Gallopoulos, E., Simoncini, V., Szeto, T., Tong, C.H.: A Quasi-Minimal Residual Variant Of The Bi-Cgstab Algorithm For Nonsymmetric Systems. SIAM Journal on Scientific Computing (1994)
19. Rixner, S., Dally, W.J., Kapasi, U.J., Mattson, P., Owens, J.D.: Memory Access Scheduling. In: 27th Annual International Symposium on Computer Architecture, Vancouver, Canada, pp. 128–138 (June 2000)
20. Lawson, C.L., Hanson, R.J., Kincaid, D., Krogh, F.T.: Basic Linear Algebra Subprograms for FORTRAN Usage. ACM Trans. Math. Soft. 5, 308–323 (1979)

On Disconnection Node Failure and Stochastic Static Resilience of P2P Communication Networks

F. Safaei[1,2], M. Fathy[1], A. Khonsari[3,2], and N. Talebanfard[4]

[1] Dept. of Computer Eng., Iran Univ. of Science and Technology, Tehran, Iran
[2] IPM School of Computer Science, Tehran, Iran
[3] Dept. of ECE, Univ. of Tehran, Tehran, Iran
[4] Faculty of Mathematical Sciences, Shahid Beheshti Univ., Tehran, Iran
{f_safaei, mahfathy}@iust.ac.ir, {safaei, ak}@ipm.ir

Abstract. There exist a large number of graph optimization problems in the literature, which arise in network design and analysis. Our objective in this paper is to highlight the disconnection probability which can arise in interconnect networks of large-scale parallel processors systems. Although traditional measures of fault-tolerance such as reliability and availability are applicable to such systems, these measures were designed mostly for mission-oriented applications or repairable systems. They fail to account for the high redundancy levels typical in peer-to-peer (P2P) communication and distributed systems. For these systems, new measures have been introduced that can evaluate the capability of a system for graceful degradation. In the design of such systems, one of the most fundamental considerations is the reliability of their interconnected networks, which can be usually characterized by connectivity of the topological structure of the network. In this paper, we analyze the problem of network disconnection in the context of large-scale P2P networks and understand how static patterns of node failure affect the resilience of such networks. Simulation results based on the network topology confirm the validity of the analytical approximation and demonstrate the localizer efficiency.

1 Introduction

Modern science, engineering and technology pose a great number of important problems whose solutions require an enormous amount of computational or processing power, and in some cases the solutions must be obtained within a small time period; for example, image processing, radar signal processing, weather tracking and prediction, control of fast and complex nuclear or chemical reactions, aerodynamic simulation and many more [1]. It is imperative that we should have very large-scale computers consisting of high-speed processors capable of executing parallel algorithms. However, the speed of processors will be pushed to the limits established by quantum mechanics. By most accounts, the next generation of supercomputers will achieve its gains by increasing the number of processing elements and the use of concurrency, rather than using faster processors. Consequently, the designers are required not only to build reliable, but also to provide various supporting software matching with them. Due to recent advances in hardware technology, especially in the advent of VLSI

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 367 – 378, 2007.
© Springer-Verlag Berlin Heidelberg 2007

circuit technology, it is now feasible to build a large and very complex interconnect networks connecting tens of thousand processing elements for P2P communication networks which are capable of executing parallel algorithms. Thus, a systematic method is very necessary to design a topological structure for such very large-scale interconnect networks.

Large-scale parallel computers, Multiprocessors System-on-Chip (MP-SoCs), multicomputers, cluster computers and P2P communication networks are considered today as a very promising approach of achieving high computational power. A basic feature for such system is that its components are connected together by physical communication links to transmit information according to some pattern. Moreover, it is undoubted that the power of a system is highly dependent upon the connection pattern of components in the system. A connection pattern of the components in a system is called an interconnect network, or network for short. Topologically, an interconnect network can essentially depict structural feature of the system. In other words, an interconnect network of a system provides logically a specific way in which all components of the system are connected. It is quite natural that an interconnect network may be modeled by a simple graph whose vertices represent components of the network and whose edges represent physical communication links, where undirected edges represent two-way communication links, and the incidence function specifies a way that components of the network are interconnected by links. Such graph is called topological structure of a network, or network topology for short. Usually the network topologies can be grouped into two categories: dynamic and static [2]. In a dynamic system, the links can be reconfigured by setting the network's active switching elements. In a static system, on the other hand, the communication links between processors are passive and reconfiguration of the system is not possible. In this paper, we are mainly interested in a static topological structure of interconnect networks.

Fault-tolerance is important to allow usage of the interconnect network in the presence of failures. Preferably, faults and changes should be tolerated in run-time without disrupting network operation. If one routing node or one link fails, the rest of the network should not be forced into a halt. Preferably, they should be allowed continuous use of the interconnect network, suffering only from the degradation of performance that the fault represents. If a routing node or a link is inserted into the network, the network should incorporate the extra resource without requiring a stop in the communication between the communicating devices.

In the design of an interconnect network, one of the most fundamental considerations is the reliability of the networks, which can be usually characterized by connectivity of the topological structure of the network. In this paper, we will briefly recall the concept of connectivity of a graph and some basic results used in the paper. Faults of some components and/or communication lines in a large-scale system are inevitable. However, the presence of faults gives rise to a large number of problems to have to be handled for some applications. Generally speaking, the solutions of these problems are difficult, as the set of faults is not known in advance. Nevertheless, they have attracted considerable research interest of the recent decade, and many nice results have been obtained [2-7]. From the preceding discussions we have known that the

connectivity of a network have played an important role in fault-tolerant analysis of interconnect networks since it can be used to measure fault-tolerance of the networks. In this paper, we will introduce one of the important problems; formalize it as standard concepts of graph theory and mathematical analysis which present some basic research results.

The article is structured as follows. Section 2 examines previous work relevant to this paper. Section 3 reviews some definitions and background of the torus and mesh topologies. In Section 4 several principles of fault-tolerance in P2P communication systems are recalled. In Section 5, we introduce our approach to analyze the probability of occurrence of a network disconnection. In Section 6, analytical approximations of this probability are confirmed through a simulation approach. The disconnection probability measures of the random graphs and the torus and mesh networks of various sizes are compared. Moreover, the influence of the disconnection probability on the reliability and performance is presented in this section. Concluding remarks and directions for future research are presented in Section 7.

2 Related Work

Resilience of graphs [8-10] and various types of deterministic networks have attracted significant attention in the research literature. A classical problem in this line of study is to understand failure conditions under which the network disconnects and/or starts to offer noticeably lower performance (such as increased routing distance) to its users. To this end, many existing models assume uniformly random edge/node failure and derive the conditions under which each user [11], certain components [12], or the entire graph [13] stay connected after the failure. Current analysis of P2P networks frequently adopts the same model of uniform, concurrent node failure and ranges from single-node isolation [11, 14] to disconnection of the entire graph [15]. However, these studies rarely discuss the scenarios under which large quantities of P2P users may simultaneously fail or accurately estimate failure probability p in practical systems [16]. The original thrust [1] in this direction focused on *static* node failure, where a fully populated network experienced simultaneous node failure with independent probability p. While analytical results on the exact probability of disconnection under static failure are currently unavailable in the literature, prior analysis suggests that P2P networks are highly resilient to node faults and can survive the failure of up to 50% of the graph without significant degradation in performance [9, 10]. A different approach is also to examine disconnection in dynamic systems [10], where users continuously join and leave the network according to some arrival/departure processes. Connectivity of generic deterministic graphs under independent node failure has also received significant attention in the literature [1-10]. In this line of work, $f(p)$ is called *residual node connectivity* and can be written as [1, 10]

$$f(p) = \sum_{i=0}^{N} C(i) p^i (1-p)^{N-i} \tag{1}$$

where $C(i)$ is the number of connected induced sub-graphs of G with exactly i nodes and all node failures are assumed to be independent. The main difficulty in using $f(p)$ is computing $C(i)$. $C(i)$ can be determined by enumeration if N is reasonably small. In the case of very large networks, it may be necessary to use approximation methods such as simulation to compute $C(i)$. A measure of the network reliability is defined as

$$R(p) = 1 - f(p) \tag{2}$$

Computation of $R(p)$ for a generic graph requires the knowledge of an NP-complete [1, 10, 17] metric $C(i)$, whose expression is unknown even for the basic hypercube network.

Najjar and Gaudiot [8], however, noticed that several non-hypercube deterministic networks frequently develop disconnection around individual nodes rather than along boundaries of large sets $C(i)$. This leads to a model for the probability, that a N-node, k-regular graph is partitioned under p-percent node failure [8]. Other approaches that study disconnection of hypercubes include; however, none of them provides a practically usable model that is both accurate and simple to evaluate. Since the links themselves are assumed to be redundant and highly reliable with error correcting code transmission, the research presented in this paper only focuses on static node failure and is a probabilistic evaluation for static resilience which alludes to the connectivity of a graph after each node is removed from the graph independently with probability p.

3 Background

3.1 The Topological Structures

The topological structure of an interconnect network can be modeled by a graph. This fact has been universally accepted and used by computer scientists and engineers. Moreover, practically it has been demonstrated that graph theory is a very powerful mathematical tool for designing and analyzing topological structure of interconnect networks. In this section, we will briefly recall some basic concepts and notations of graph theory used in this paper as well as the corresponding backgrounds of networks.

The mesh and grid networks

The mesh networks frequently appear in various applications of networks. The topological structure of a mesh network is defined as the Cartesian product $P_l \times P_m$ of undirected paths P_l and P_m, denoted by $G(l,m)$ [1]. Sometimes, $G(l,m)$ is called an $l \times m$ mesh in the literature. It is quite natural to generalize the mesh network to the grid network. The *n*-dimensional grid network, denoted by $G(m_1,\ldots,m_n)$, is defined as $P_{m_1} \times \cdots \times P_{m_n}$, where P_{m_i} is an undirected path of length m_i for each $i = 1,2,\ldots,n$.

The torus network

An n-dimensional torus consists of $k_{n-1} \times k_{n-2} \times \ldots \times k_i \times \ldots \times k_0$ nodes, k_i nodes along each dimension i, where $k_i \geq 2$. Each node X is denoted by $(x_{n-1}, x_{n-2}, \ldots, x_i, \ldots, x_0)$, where $0 \leq x_i \leq k_i - 1$. A node $(x_{n-1}, x_{n-2}, \ldots, x_i, \ldots, x_0)$ is connected to nodes $\left(x_{n-1}, x_{n-2}, \ldots, x_{i+1}, (x_i \pm 1) \bmod k_i, x_{i-1}, \ldots, x_0\right)$, for all $0 \leq i \leq n$.

4 Fault-Tolerance and Connectivity of P2P Networks

In designing or selecting a topological structure of interconnect network for a system, one fundamental consideration is the fault-tolerance. There are two types of faults in a system: hardware failures and software errors. Most hardware failures have physical causes such as component wear and electromagnetic interference. Software errors are primarily due to design errors. The former are able to be modeled mathematically and the latter are much more difficult to model. Avizienis [18] has succinctly defined fault-tolerance in a computer system as "the ability of a system to execute specified algorithm correctly regardless of hardware failures and program errors". However, Avizienis' definition of fault-tolerance is by no means easy to apply to parallel computer systems. The performance of a fault-tolerant system should include two aspects: computational efficiency and reliability. When a component or link fails, its duties must be taken over by other fault-free components or links of the system. If those components or links are used to improve only the reliability of the system and not affect its computing performance, they are called redundant. On the other hand, computation speed can be influenced under the presence of failures. For example, it may be necessary to devote a substantial amount of processing time to the execution of test and maintenance programs. Furthermore, the occurrence of failures may cause a reduction in processing performance. The ability of a faulty system to continue to operate at an acceptable but lower level of performance is referred to as graceful degradation. Following Esfahanian [19], a system is said to be fault-tolerant if it can remain functional in the presence of failures. It is, however, the topological requirements set by the application environments that essentially determine when a system is considered to be functional.

Two basic functionality criteria have received much attention. According to one of these criteria, a system is considered functional as long as a certain topological structure is logically contained in the network. This is the problem of embedding an architecture into another. The approach in achieving fault-tolerance, when this functionality criterion is used, is to employ system wide redundancy and reconfiguration. This criterion was first formulated by Hayes [20].

The second functionality criterion considers a communication network functional as long as there is non-faulty communication path between each pair of non-faulty components. In other words, the topological structure of the network remains connected in the presence of certain failures. Throughout this paper, the terms "fault-tolerance" will only refer to this situation. This fault-tolerant model is particularly applicable to a large-scale parallel computer that is to execute concurrent algorithms

whose performance is mostly insensitive to infrequent changes in the topological structure of the system. Such a system generally permits graceful degradation. Several authors have proposed a variety of measures of fault-tolerance. These measures are either deterministic or probabilistic [1, 8-10]. Researchers have mainly used graph-theoretic concepts to develop deterministic measures of fault-tolerance. The notion of connectivity is the key graph-theoretic concept employed. If G is the topological structure of an interconnect network, then the connectivity $\kappa(G)$ of G becomes the smallest number of components whose breakdown would jeopardize communication in the network, in other words, to remain connected the network can simultaneously tolerate $\kappa(G)-1$ component failures. Thus, the higher connectivity leads to more reliable network. Probabilistic measures of fault-tolerance seem to model the real world better than their deterministic counterparts. Most network performance evaluation studies have been conducted by means of software simulation. Studying the relative performance merits of network algorithms using simulation techniques is, however, limited by the excessive computation times required to run large simulations. Analytical modeling, in contrast, offers a cost-effective and versatile tool to carry out such a study typically requiring a far lower computational load. The research presented in this paper is a probabilistic evaluation of network fault-tolerance based on evaluating the disconnection probability at the i-th element failure.

5 Calculating the Probability of Disconnection

In addition to the loss of computational power, the node failure in a large-scale P2P system has the effect of modifying the system topology by removal of a communication node and its corresponding links. The topology modification would eventually lead to the disconnection of the system. A system is said to be disconnected when its graph topology has two or more disjoint components [1].

In order to calculate the reliability of such system, the probability of a disconnection is thus a central parameter. When a failure isolates a healthy subset from the rest of the computation, it can be said that the system has irrecoverably failed. In this section, we propose a probabilistic measure of network fault-tolerance expressed as the probability of occurrence of a disconnection. The disconnection problem presented and analyzed here is not specific to the studied cases, i.e. torus or mesh, and the results can be extended to any other topologies.

Let $G=(V,E)$ be a connected undirected graph, and n be the number of nodes of G. Let $v \in V(G)$ be a node in G. We use the symbol $N(v)$ to denote the set of neighbors of v in G, and let $d(v)=|N(v)|$ be the cardinality of the set $N(v)$. We also denote by G' the remaining graph after the node failures. In this section, we capture the probability that G' remains connected under independent p-percent static node failure. Let Y be the number of remaining nodes, and X_i be the number of connected components of size i in G'. Moreover, let C and DC denote the events that G' is connected and disconnected, respectively. To deal with the problem, we first notice that the following relations hold.

$$\begin{aligned}\Pr[DC] &= \Pr\Big[\sum_{i=1}^{\lfloor Y/2 \rfloor} X_i \geq 1\Big] \\ &\leq E\Big[\sum_{i=1}^{\lfloor Y/2 \rfloor} X_i\Big] \\ &= \sum_{i=1}^{\lfloor Y/2 \rfloor} E[X_i] \\ &\leq \sum_{i=1}^{\lfloor Y/2 \rfloor} \sum_{S \subseteq V(G), |S|=i} (1-p)^i \prod_{v \in S} p^{|N(v) \setminus S|}\end{aligned} \tag{3}$$

where S is a subset of nodes in the network and the operator "\" signifies those subsets of $N(v)$ that do not contain S, i.e., $N(v) \setminus S = \Big\{ v \in N(v) : v \notin S \Big\}$.

We do so with the following steps. First, suppose that $\sum_{i=1}^{\lfloor Y/2 \rfloor} X_i \geq 1$. This means that $X_i \geq 1$ for some $1 \leq i \leq \lfloor Y/2 \rfloor$. But since X_i does not contain all the nodes, G' is not connected. On the other hand, assume that G' is not connected. Let $C_1, \ldots, C_k$ denote the connected components in G'. It immediately follows that

$$|C_i| \leq \frac{1}{k} \sum_{i=1}^{k} |C_i| = \frac{Y}{k}, \text{ for some } 1 \leq i \leq k \tag{4}$$

But since G' is disconnected, we have $k \geq 2$ and then there is i such that $|C_i| \leq \lfloor Y/k \rfloor$, and therefore the first equality holds. Applying Markov's inequality $P(X \geq 1) \leq E[X]$, we directly obtain the first inequality. The second equality, however, follows from the linearity of expectation, and the last inequality holds, since S is a connected component, there exists no $v \in S$ such that has a neighbor outside S. For convenience, let

$$\phi(G, x) = \sum_{i=1}^{\lfloor x/2 \rfloor} \sum_{S \subseteq V(G), |S|=i} (1-p)^i \prod_{v \in S} p^{|N(v) \setminus S|} \tag{5}$$

This in turn can lead to the following model

$$\Pr[C] \geq 1 - \phi(G, Y) \tag{6}$$

Since $E[Y] = (1-p)n$, the Eq. (5), on average, can be approximated by

$$\Pr[C] \approx 1 - \phi(G, (1-p)n) \tag{7}$$

Our next approach to compute $\Pr[C]$ is based on a result of Erdös and Rényi [21], which states that G' is almost surely connected if and only if it has no isolated nodes. Let Z_v be the indicator random variable denoting that v stays alive and becomes isolated after node failures. Therefore, the number of isolated vertices in G' could be computed as $Z = \sum_{v \in V(G)} Z_v$. We evidently have

$$\Pr[Z_v = 1] = (1-p)\,p^{d(v)} \tag{8}$$

Thus, the expected number of isolated nodes is given by

$$E[Z] = \sum_{v\in V} E[Z_v] = \sum_{v\in V} (1-p)\,p^{d(v)} = (1-p)\sum_{v\in V} p^{d(v)} \tag{9}$$

Since Z is a binomial random variable, we can apply a Poisson approximation [9] and thus we have

$$\Pr[C] \approx \Pr[Z = 0] \approx e^{-(1-p)\sum_{v\in V} p^{d(v)}} \tag{10}$$

Therefore, assuming G is k-regular, we can write

$$\Pr[C] \approx e^{-n(1-p)p^k} \tag{11}$$

Let us now generalize the notion of connectivity which may arise in real life problems. We say that G is k-connected if and only if it has at most k connected components. Notice that, 1-connectivity is the equivalent to the classical notion of connectivity. Using the result of Erdös and Rényi [21], we can say that G' is almost surely k-connected if and only if it has at most $k-1$ isolated nodes. Let C_k denote the event that G' is k-connected. We thus have

$$\Pr[C_k] \approx \Pr[Z \le k-1] \approx e^{-(1-p)n\sum_{i=0}^{k-1}\frac{(1-p)^i n^i}{i!}} \tag{12}$$

6 Simulation Results

An experimental approach is necessary to verify the analytical approximation to which mathematical analysis led. A program has been developed which simulates the failure of nodes and the subsequent removal of the corresponding links. It also checks for eventual disconnections in the graph. The objective of the simulation is to measure the values of the probability of disconnection for different values of faulty nodes in random graphs and the torus and mesh topologies. For every run, the simulator keeps statistics of the following data:

- The values of i for which the disconnection occurred.
- The average number of healthy links remaining in the network after each failure.

Fig. 1 shows the algorithm was used in the simulation approach. In the simulation approach, each number of iterations was approximately set to 10,000. As first results, we present the probability of occurrences of disconnection for different values of network size and faulty nodes in networks. Fig. 2 depicts plots of the disconnection probability distribution function as a function of the percentage of failed nodes for

10×10, 12×12×12 mesh and torus networks and random graphs with N = 256, and 1024 nodes. These measures have been obtained both by analytical and by simulation approaches. It can be seen that larger networks have the potential to survive a larger percentage of node failures. This is a direct consequence of their higher connectivity. The figure also reveals that no disconnection can occur when the number of failures is less than the number of neighbors of the subset to be disconnected. It eloquently shows that increasing the network size, N, the probability of disconnection becomes significant.

```
01 Subroutine TraverseNode (current node)
02  begin
03    if current node is healthy then
04       begin
05         Increment number of traversed nodes
06        Proceed traversing all of its neighboring nodes
07       end
08  end
09 Algorithm: Calculation the probability of disconnection in P2P networks
10 Make the topology
11   for i ← 0 to 100 with step do /* step is the incremental value of failure rate */
12     begin
13       for i ← 0 to Number of iterations do
14         begin
15           Build a random faulty list with respect to the value of faulty nodes
16           Make a graph with respect to the topology type and faulty list
17           Return a healthy node at random
18           TraverseNode (selected healthy node)
19                 if number of traversed nodes + number of faulty nodes = N then
20                   Increment number of connected nodes
21          end
22       Record (connected value / number of iterations) ×100% into the RecordTable
23 /*RecordTable is a table whose elements are total percentage of graph connectivity
24   over the failure rate*/
25 end
```

Fig. 1. Pseudocode that used in the simulation program to compute the probability of disconnection in the P2P networks

However, after a specific point (which is referred to as the failure point) for larger values of failed nodes this value reaches null because the probability of a disconnection of a sub-network is proportional to the number of possible subsets, which can be disconnected. Furthermore, it is evident that the torus networks have a higher connectivity and would thus be less susceptible to disconnections than the mesh. Comparing the simulation results and the analytical approximation obtained from Eqs. (3) ~ (12) reveals that for all practical purposes and for larger networks, the proposed analytical model is a precise approximation as evidenced by the simulation experiments.

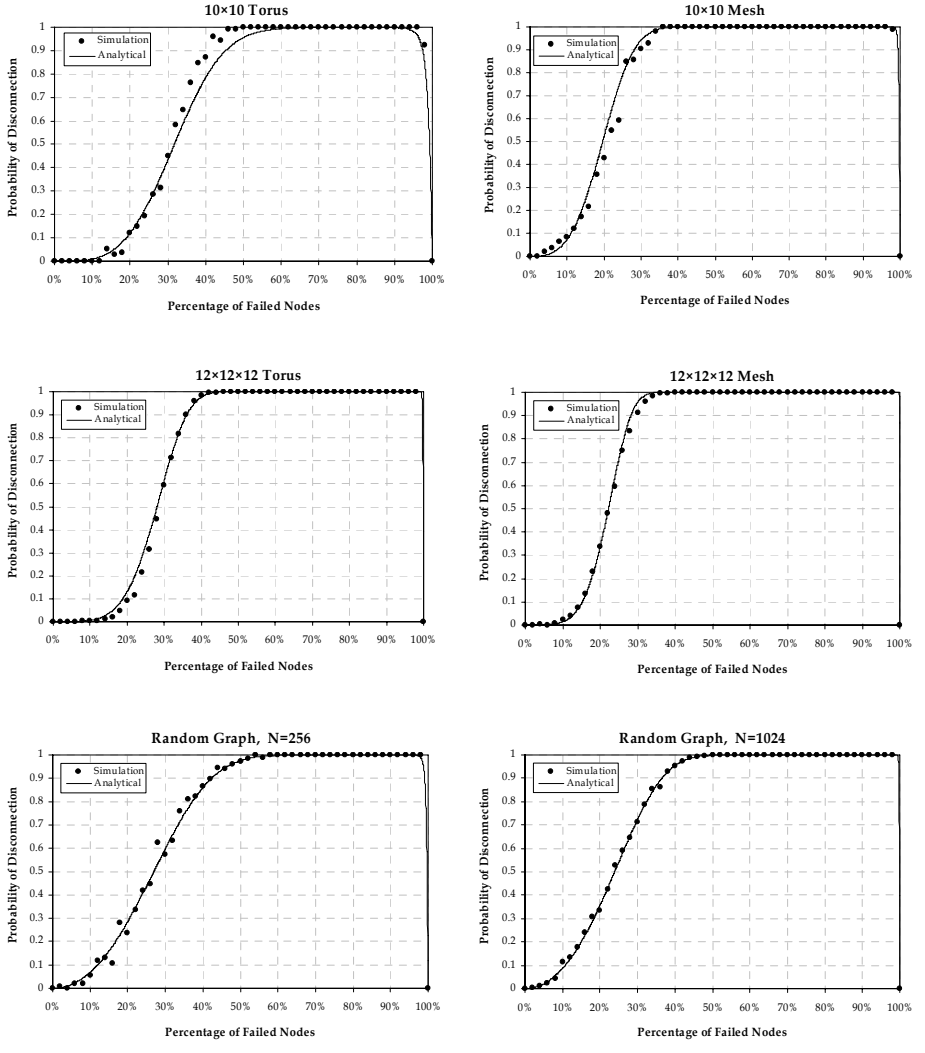


Fig. 2. Probability of disconnection calculated by the analytical model against those obtained from simulation for 10×10, 12×12×12 torus and mesh networks and random graphs with N = 256, and 1024 nodes

7 Conclusions

In this paper, we have investigated the reliability of P2P communication systems. The issue of fault-tolerance has been studied to focus on the important disconnection probability measure. We have confirmed ourselves to a set of random graphs and mesh and torus networks of different dimensionality. It has been shown that the network size has a great impact on the disconnection probability. Larger systems are

more vulnerable to the number of faulty components. The probability of disconnection has been calculated through a novel mathematical analysis inference. Moreover, we have introduced the network reliability measure, $R(p)$, which denotes the number of nodes that network still remains connected with a probability of $1 - f(p)$. This measure is constructive in the sense that it introduces an upper bound on the number of allowable degradation states in a gracefully degradable non-reconfigurable fault-tolerant system. The disconnection problem presented and analyzed in this paper is not specific to the mesh or torus network, but can be encountered in any other system topology. Future work includes driving a model for dynamic resilience where users depart the system after spending random amounts of time online, and extending the current model to degree-irregular networks, and constructing more resilient P2P communication networks.

References

1. Xu, J.: Topological structure and analysis of interconnection networks. Kluwer Academic Publishers, Dordrecht (2001)
2. Suh, Y.J., et al.: Software-based rerouting for fault-tolerant pipelined communication. IEEE Trans. On Parallel and Distributed Systems 11(3), 193–211 (2000)
3. Go'mez, M.E., et al.: A Routing Methodology for Achieving Fault Tolerance in Direct Networks. IEEE Trans. On Comput. 55(4), 400–415 (2006)
4. Mejia, A., et al.: Segment-Based Routing: An Efficient Fault-Tolerant Routing Algorithm for Meshes and Tori. In: IEEE International Parallel & Distributed Processing Symposium (IPDPS) (2006)
5. Montanana, J.M., et al.: A Transition-Based Fault-Tolerant Routing Methodology for InfiniBand Networks. In: IEEE International Parallel & Distributed Processing Symposium (IPDPS) (2004)
6. Boppana, R.V., Chalasani, S.: Fault-Tolerant Wormhole Routing Algorithms for Mesh Networks. IEEE Trans. Computers 44(7), 848–864 (1995)
7. Chakravorty, S., Kale, L. V.: A Fault Tolerant Protocol for Massively Parallel Systems, IEEE International Parallel & Distributed Processing Symposium (IPDPS) (2004)
8. Najjar, W., Gaudiot, J.L: Network resilience: A measure of network fault-tolerance. IEEE Transactions on Computers 39(2), 174–181 (1990)
9. Leonard, D., et al.: On static and dynamic partitioning behavior of large-scale networks, ICNP05, pp. 345–357 (2005)
10. Leonard, D., et al.: On lifetime-based node failure and stochastic resilience of decentralized peer-to-peer networks. In: SIGMETRICS, pp. 26–37 (2005)
11. Stoica, I., et al.: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In: Proceedings of ACM SIGCOMM'01, San Diego (2001)
12. Bollobas, B.: Random graphs. Cambridge Univ. Press, Cambridge (2001)
13. Gummadi, K.: The impact of DHT routing geometry on resilience and proximity. In: ACM SIGCOMM, pp. 381–394 (2003)
14. Kaashoek, F., Karger, D.R.: Koorde: A simple degree-optimal distributed hash table. In: Kaashoek, M.F., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735, Springer, Heidelberg (2003)

15. Massoulié, L., Ganesh, A.J., Kermarrec, A.M.: Network Awareness and Failure Resilience in Self-Organizing Overlay Networks. In: IEEE Symposium on Reliable and Distributed Systems, Florence (2003)
16. Fiat, A., Saia, J.: Censorship Resistant Peer-to-Peer Content Addressable Networks. In: Proceedings of Symposium on Discrete Algorithms, ACM-SIAM SODA (2002)
17. Sutner, K., Satyanarayana, A., Suffel, C.: The complexity of the residual node connectedness reliability problem. SIAM Journal on Computing 20(1), 149–155 (1991)
18. Avezienis, A.: Fault-tolerant computing- an overview. IEEE Transactions on Computers 4, 5–8 (1971)
19. Zimmerman, G.W., Esfahanian, A.H.: A New Approach to System-Level Fault-Tolerance in Message-Passing Multicomputers. In: Great Lakes Computer Science Conference 38(11), 357–363 (1989)
20. Hayes, J.P.: A Graph Model for Fault-Tolerant Computing Systems. IEEE Transactions on Computers 25(9), 875–884 (1976)
21. Erdös, P., Rényi, A.: On the evolution of random graphs. Publications of the Math. Inst. of the Hungarian Academy of Sci. 5, 17–61 (1960)

An Efficient Sequence Alignment Algorithm on a LARPBS

David Semé and Sidney Youlou

LaRIA : Laboratoire de Recherche en Informatique d'Amiens
Université de Picardie Jules Verne
CURI, 5 rue du Moulin Neuf, 80000 Amiens, France
{David.Seme,Sidney.Youlou}@u-picardie.fr

Abstract. In this paper, we give a parallel algorithm solving the sequence alignment problem on a LARPBS, one of the recently proposed parallel model based on optical bus. The complexity of our algorithm solving the sequence alignment problem is O(m) using n processors (where m and n are respectively the sizes of the shortest and the longest sequences).

1 Introduction

The last years saw a growing interest in computational biology, an area that is at the crosspoint of computer science and biology. This emerging field has rapidly become a major field as many articles have been devoted to it. It consists in using computer science for designing efficient algorithms in order to solve problems on biological systems. Most of the problems are now well known as they have received a lot of attention from researchers. They mostly deal with sequences analysis and include alignment problem, pattern matching [11,13], longest common and longest increasing subsequence [10,12], closest substring problem and many others. Among theses problems, the sequence alignment problem may be one of the most studied.

In its pairwise version, it consists in finding an arrangement of two biological sequences in order to measure their degree of similarity. It is solved in O(mn) time sequentially with two strings of length m and n [2]. Different approaches exist to solve this problem. Some of them use the notion of distance. A distance measure is a function that associates a numerical value with a pair of sequences with the idea that the larger is the distance, the smaller is the similarity. For example, the Hamming distance is a powerful tool used in pattern matching algorithms like the approximate string matching problem with k-mismatches. Yet, for alignment problems, it appears to be not flexible enough. The Edit distance or Levenshtein distance [6] is more appropriate for this purpose as it takes in consideration edit operations. It is defined as the minimum number of character insertions, deletions and substitutions necessary to make two strings equal. There are many extensions of the Edit distance such as the the Smith-Waterman algorithm [14] and the Neddleman-Wunsch algorithm [9]. A straightforward application of the edit distance is the approximate string matching problem with k-differences in which an edit operation reveals a difference and given an integer k, the edit distance of the pattern and an occurrence of the pattern is at most k.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 379–387, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Parallel algorithms exist for these problems on various models [5,1,8,4]. The difficulty of the parallelization of this problem is that its sequential solution is based on a recurrence formula which include a strong data dependency. Recently, on the LARPBS, Chen and al. proposed in [1] an efficient algorithm for the sequence alignment problem. To overcome the data dependency issue, they use a parallel prefix-maximum algorithm which computation is $O(d)$ where d is the number of binary bits of an integer. As a result, the total complexity is $O(md)$ time if the number of processors is n.

Our new approach reduces the complexity to $O(m)$ as we investigate the nature of the data dependency and present a new formula for computing the edit distance.

The rest of the paper is organized as follows. Section 2 gives a description of the LARPBS model. In section 3, we present our parallel solution for the sequence alignment problem. The conclusion ends the paper.

2 The LARPBS Model

The LARPBS model is presented by Pan and Li in [7]. It is based on the ideas of using pipelined bus systems and processor array reconfiguration. In such model, unlike in its electrical counterpart, several messages can circulate at the same time on the bus in a pipelined fashion. The communication time for the LARPBS model is measured based on the number of bus cycles used. A bus cycle for this model is defined as the end-to-end propagation delay on the bus which is the time taken for an optical signal to propagate through the entire bus.

The advantages of using waveguides are shown as follows: unidirectional propagation and predictable delay per unit length, in addition to the high propagation speed of light. These properties enable synchronized concurrent access of an optical bus in a pipelined fashion.

The optical bus has three waveguides. The message waveguide is used for sending data and the select and reference waveguides are used for sending address information. Each processor is connected to the bus through to directional couplers, on for transmitting and the other for receiving. Hence, the bus is divided into two segments, the transmitting segment and the receiving segment. The receiving segments of the reference and message waveguides have an extra segment of fiber between every pair of consecutive processors. This segment of fiber is used to introduce one unit propagation delay in these two waveguides. Hence the propagation delays on the receiving segments of the reference and select waveguides are not the same. Finally, a conditional delay between every pair of consecutive processors is placed on the transmitting segment segments of the select waveguide. The conditional delays can be implemented using 2×2 optical switches. Each switch can be set by a local processor to two different states: straight or cross (see Figure 1).

To add reconfiguration facility on the bus, six more optical switches are inserted on each segment (*i.e.* between every pair of consecutive processors) of the bus. Three optical switches on each transmitting and receiving segment (one

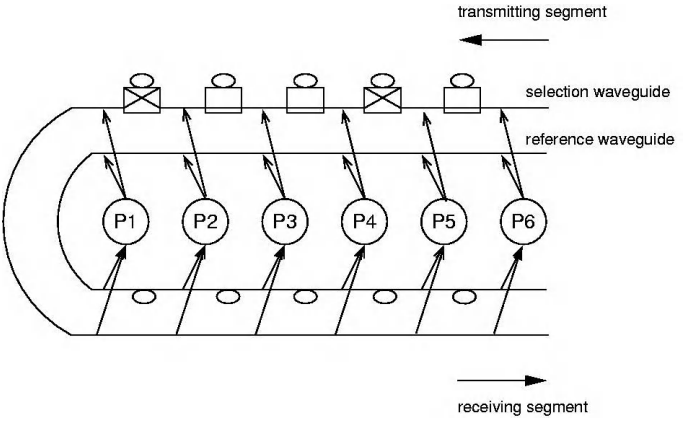


Fig. 1. The LARPBS model. Only the select and reference waveguide are represented. The message waveguide is similar to the reference waveguide. P_1 and P_4 have their switches set to cross. P_2, P_3, P_5 have their switches set to straight.

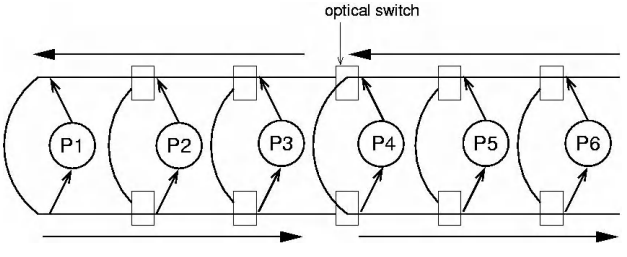


Fig. 2. A LARPBS of size 6 with two subarrays

per waveguide). If all these switches are set to straight, the bus system operates as a regular pipelined bus system connecting all the processors. However, if the switches at processors P_i are set to cross, the bus is split into two separate buses, one connecting processors $P_1P_2...P_{i-1}$ and the other connecting processors $P_iP_{i+1}...P_n$ as described in figure 2 (only one waveguide is shown). These two sets of processors can independently work as two LARPBS.

The addressing is done using the coincident pulse technique [3]. If processor P_i wants to send a message to processor P_j, P_i transmits a message frame on the message waveguide and a pair of select and reference pulses on the select and reference waveguides. The technique requires to use accurately the difference of delays between the select and reference waveguides in order to make the pulses coincide at processor P_j. With such method, it is possible to route messages or to broadcast messages on the optical bus.

Nevertheless when multiple messages arrive at the same processor in the same bus cycle, only the first message is received, the others are automatically ignored. It is due to the fact that an electrical processor cannot match the transmission speed of an optical bus [7].

3 The Sequence Alignment Algorithm

3.1 Basic Backgrounds

Given an alphabet $\sum$, an element $\sum^*$ is called a string (or a sequence). Let A and B be two strings, A=$a_1 a_2..a_m$and B=$b_1 b_2..b_n$, the length of A (respectively B) is m (respectively n).

Given a string P=$p_1 p_2..p_k$ of length k, let us denote P[i] its i^{th} element ($0<i\leq k$) and P[$i..j$] its factor $p_i p_{i+1}..p_j$ ($0<i<j\leq k$).

The *Hamming* distance is a measure of distance between two strings of same length expressed by the number of characters that need to be changed to obtain one from the other, for example *ctgga* and *caggc* have a Hamming distance of 2.

This distance is relevant for some string matching problems like the exact string matching problem where the goal is to find the occurrences of the pattern. Here, the Hamming distance between the pattern and its occurrence must be 0. Yet, searching for exact occurrences is not necessarily a good thing. Biologists sometimes prefer to retrieve approximate occurrences and some problems deal with that like the approximate string matching with k-mismatches. In this situation, the Hamming distance of the pattern and its occurrences must be at most k.

The major inconvenient with the last problem is that in some cases it appears to be too restrictive.

First, to compute the Hamming distance, the two strings must have the same length. For example, if we search for the pattern *tagc* in *atgtatgc* with at most 1 difference, the Hamming distance of the pattern and all the subwords of the reference string is at least 2. But now, if we allow edit operations such as deletion, insertion and substitution then it becomes 1. Then, we obtain a more accurate distance measure called the *edit distance*.

The *edit distance* (or Levenshtein distance) of A and B is defined as the minimum number of mutations required to change A into B, where a mutation is one of:

1. change a letter
2. insert a letter
3. delete a letter

In the weighted version of the problem, an insertion cost *ins*, a deletion costs *del*, a substitution costs *sub*, and a match costs 0.

This problem is solved in O(mn) time sequentially. The goal is to fill a matrix $m\times n$ with the following recurrence formula:

$$R[i,j] = \begin{cases} i & if\ j = 0 \\ j & if\ j = 0 \\ R[i-1, j-1] & if\ A[i] = B[j] \\ min(R[i,j-1] + ins, R[i-1,j] + del, R[i-1,j-1] + sub) & if\ A[i] \neq B[j] \end{cases}$$

The last cell of the table gives the score. The smaller is the score, the smaller is the difference between A and B.

Commonly in computer science, the operations have the same weight in order to not discriminate one of the permitted operations and the costs *ins*, *del*, and *sub* are set to 1.

With this distance measure, we can solve the approximate string matching problem with k-differences where the goal is to find approximate occurrences of the pattern where the edit distance between the pattern and the occurrences is at most k. The previous recurrence formula slightly changes, the second rule is: 0 if j=0.

3.2 The New Formula

The problem with the parallelization of the previous formula is that there is the strong data dependency. The computation of a cell only requires to know the information in 3 cells. But given a n-processor LARPBS and in order to propose an optimal algorithm, we need to compute each row in O(1) time.

Suppose that the row i-1 have been computed then to compute row i, R[i,j] knows R[i-1,j-1] and R[i,j-1] but can only knows R[i,j-1] once it has been computed. In the the same time, a cell where A[i]=B[i] can be filled straightly because it only needs R[i-j,j-1].

Thanks to the reconfiguration ability of the LARPBS, we can use this last information to fill a row in O(1).

$$R[i,j] = \begin{cases} i & if\ j = 0 \\ j & if\ j = 0 \\ R[i-1, j-1] & if\ A[i] = B[j] \\ min(R[i,0] + j * del, R[i,k] + (j-k) * del, R[i-1,j] + ins, R[i-1,j-1] + sub) & if\ A[i] \neq B[j] \\ where\ k = max\, k \in (1,..,j-1)/A[i] = B[k] & \end{cases}$$

Proof. The difference between this formula and the classical one is the fourth rule. Let us show that both formulas are equivalent.

Let A and B be two sequences of size m and n.

First case, Suppose there is no match on the i^{th} row, *i.e.* A[i]≠B[j], j=1,..,n.

Then with the classic formula R[i,0]=i and R[i,1]=min(R[i,0]+*del*, R[i-1,0]+*sub*, R[i-1,1]+*ins*).

R[i,0]+*del*=R[i,0]+1**del*.

Now suppose that the operations performed on the k-1 first cells are deletions.

It means that for j=1,..,k-1: R[i,j]=R[i,j-1]+*del*. So for j=1,..,k-1 the difference between any two consecutive cells is *del* and then for j=1..,k-1 R[i,j]= R[i,0]+j**del* because R[i,j-1]+*del*=R[i,0]+j**del*.

Let us see the case for the remaining cells on the row.

We have R[i,k-1]+*del*<R[i-1,k]+*sub* or R[i,k]+*del*<R[i-1,k+1]+*ins* *i.e.* the substitution or the insertion cost less than the deletion. The consequence is that in order to compute R[i,k+1] we have R[i,k]+*del*<R[i,0]+(k+1)**del*.

Property 1. On a row where A[i]≠B[j], j=1,..,n, if R[i,j]+*del*<R[i,0]+(j+1)**del* then R[i-1,j]+*sub*≤R[i,j]+*del* or R[i-1,j+1]+*ins*≤R[i,j]+*del*.

This property implies two things.

1: The cost of a deletion can be the only smallest among the three costs only until a substitution or a insertion appears. After that it can be at most equal to the two others.

2: We can compute the remaining cells only with those on the upper row.

We prove this by showing that on the same row, in terms of costs, an operation of insertion or substitution can allows be followed by an operation of substitution or insertion.

R[i,j] is the result of an insertion: R[i,j]=R[i-1,j]+ins. It means that R[i-1,j-1]+sub≥R[i-1,j]+ins and R[i,j-1]+del≥ R[i-1,j]+ins.

What is the value of R[i,j+1]?

We know that R[i,j+1]=min(R[i,j]+del, R[i-1,j]+sub, R[i-1,j+1]+ins).

R[i,j]=R[i-1,j]+ins then R[i,j]+del=R[i-1,j]+ins+del.

It is easy to see that between any two consecutive cells on a same row, the different is at most del, i.e. R[a,b+1]≤R[a,b]+del.

So R[i-1,j+1]+ins≤R[i-1,j]+del+ins.

We have just shown that the operation of insertion is in the worst case equal to the operation of deletion. So, in terms of costs, we can only consider the insertion and the substitution operations.

Let us see the case of a substitution.

R[i,j] is now the result of a substitution: R[i,j]=R[i-1,j-1]+sub.

We want to compute R[i,j+1].

We know that R[i,j+1]=min(R[i,j]+del, R[i-1,j]+sub, R[i-1,j+1]+ins).

R[i,j]+del=R[i-1,j-1]+sub+del.

R[i-1,j]+sub≤R[i-1,j-1]+del+sub.

We have shown that performing a substitution at that point costs at most as much as performing a deletion. We then can only consider the costs of the insertion and substitution operations.

At this point we have proven the equivalence of the two formulas in the case where there is no matching characters on the same row.

Second case: let us see what happen when a match occurs.

Suppose that A[i]=B[k] for k∈{1,..,n}

We know that R[i,k]=R[i-1,k-1]. It is clear that in order to fill that cell and the rest, it is not necessary to know about the previous ones on the same row. We then have two independent sets R[i,0] to R[i,k-1] and R[i,k] to R[i,m].

We know that the first set of cells can be filled with the new formula. For the second set, it is the same situation except that the starting point is now the cell where the match occurred. We still have the fact that an operation of insertion or substitution can be followed in terms of cost by an insertion or a substitution.

The property implies that deletions can be performed only before insertions or substitutions appear. So, as long as this operation is chosen among the three ones, we can predict the the cost: for j>k, R[i,j]+(j-k)*del.

It is clear that we can generalize the formula for the cases where several matches occur on the same row.

This ends the proof.

3.3 The Sequences Alignment Algorithm on a LARPBS

In this part, we describe an efficient algorithm for the pairwise sequence alignment problem. We take advantage of the fact that broadcasting a message is O(1) bus cycle time and of the reconfiguration ability. We built row by row, the table of the edit distance.

Given two strings A=$a_1 a_2..a_m$ and B=$b_1 b_2..b_n$ of size respectively m and n, we need a m+1 processor LARPBS. Processor P_i ($1 \leq i \leq n$) contains B[i] in its variable *element* and an extra processor called source contains A.

Each processor also contains an array R[0..m] of size m+1 that represents a column of the table, a boolean *equal,* an integer i that represent the i^{th} cycle and a variable *id* which represent its number.

During the initialization phase, *equal* is set to false, i is set to 1 and for all the processors P_i, R[0] is set to i.

An algorithm on the LARPBS model is divided into cycles composed of an emission phase, a reception phase and a processing phase. Three cycles with different programs (cycle 1, cycle 2 and cycle 3) are required to process an element of A and fill a row of the table.

The primitive send() is for sending a message and the primitive get() is for receiving a message. get(*var*) means that the received message is set in the variable *var*.

cycle 1 : The processors receive A[i] and if A[i] is equal to the element of B they carry, then the boolean *equal* is set to true.

Algorithm 1. cycle 1

```
0  emission phase:
1   the source sends A[i] to processors P_{i=1,..,n}

2  reception phase:
3  get(element_received)

4  processing phase:
5  if (element_received=element)
6    then equal←true
7  endif
```

cycle 2 : All the processors send R[i-1] to their successor and those which are such that *equal*=true can apply the middle rule (line 17). Then they can partition the array according to the third rule by setting their optical switches to cross. We obtain several subarrays that are such that only the first processor has its variable *equal* set to true. (Lines 13 to 15 are for processor P_1 because it cannot get a message from a predecessor)

Algorithm 2. cycle 2

```
8  emission phase:
9  send(R[i-1]) to processor P_{id+1<n}

10 reception phase:
11 get(upper_left)

12 processing phase:
13 if (id=1)
14   then upper_left←i-1
15  endif
16 if (equal=true)
17   then R[i]←upper_left
18        segment_array
19 endif
```

cycle 3 : The first processor of each subarray broadcasts its variable R[i] and its number to the other processors on the subarray. These processors can apply the third rule (line 33). Then the optical switches are set to straight and the variables i and false a reinitialized for the element of A, A[i+1].

Algorithm 3. cycle 3

```
20 emission phase:
21 if (equal=true)
22   then send(R[i], id) to processors P_{j>id}
23 endif

24 reception phase:
25 get(value_received)
26 get(id_received)

27 processing phase:
28 if no message were received during this cycle
29   then left←i+id*del
30   else left=value_received+(id-id_received)*del
31 endif
32 if (equal=false)
33   then R[i]←min(left, R[i-1]+ins, upper_left+sub)
34   else connect_array
35 endif
34 i←i+1
35 equal←false
```

4 Complexity Analysis

Algorithm 1 and Algorithm 3 consist in a broadcast operation, one processor sends a message to all the others. This operation is finished in constant bus

cycle time on a LARPBS. Algorithm 2 is a one-to-one communication operation that is also performed in constant time on this parallel model.

5 Conclusion

In this paper, we give a parallel algorithm solving sequence alignment problem on a LARPBS, one of the recently proposed parallel model based on optical bus. The complexity of our algorithm solving the sequence alignment problem is $O(m)$ using n processors (where m and n are respectively the sizes of the shortest and the longest sequences). Note that this algorithm can be applied to solve the approximate string matching problem with k-mismatches.

References

1. Chen, L., Juan, C., Pan, Y.: Fast scable algorithm on LARPBS for sequence alignment. In: ISPA Workshops, pp. 176–185 (2005)
2. Gotoh, O.: An improved algorithm for matching biological sequences. Journal of Molecular Biology 162(3), 705–708 (1982)
3. Guo, Z., Melhem, R., Hall, R., Chiarulli, D., Levitan, S.: Array processors with pipelined optical busses. Journal of Parallel and Distributed Computing 12, 269–282 (1991)
4. Hoang, D.T., Lopresti, D.P.: FPGA implemenatation of systolic sequence alignment, pp. 183–191 (1992)
5. Lecroq, T., Myoupo, J.F., Semé, D.: A one-phase parallel algorithm for the sequence alignment problem. Parallel Processing Letters 8(4), 515–526 (1998)
6. Levenstein, V.I.: Binary codes capable of correcting spurious insertions and deletions of ones. problems of Information Transmission 1, 8–17 (1965)
7. Li, K., Pan, Y.: Linear array with a reconfigurable pipelined bus system: Concepts and applications. Information sciences An International Journal 11, 237–258 (1998)
8. Lopes, H.S., Moritz, G.L.: A distributed approach for a multiple sequence alignment algorithm using a parallel virtual machine. In: Proceedings of the 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Changai,China (2005)
9. Needleman, S.B., Wunsch, G.L.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48, 444–453 (1970)
10. Semé, D., Youlou, S.: Computing the longest common subsequence on a linear array with reconfigurable pipelined bus system. In: PDCS, pp. 49–54 (2005)
11. Semé, D., Youlou, S.: Parallel algorithm for string matching problems on a linear array with reconfigurable pipelined bus system.In: PDCS, pp.55–60 (2005)
12. Semé, D., Youlou, S.: Parallel solutions of the longest increasing subsequence problem using pipelined optical bus system. In: PDPTA, pp. 617–622 (2005)
13. Semé, D., Youlou, S.: Repetitons detection on a linear array with reconfigurable pipelined bus system. International Journal of Parallel, Emergent and Distributed Systems (2007) (to appear)
14. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequent. Journal of molecular 147, 195–197 (1981)

An Effective Unconditionally Stable Algorithm for Dispersive Finite Difference Time Domain Simulations

Omar Ramadan

Department of Computer Engineering,
Eastern Mediterranean University,
Gazimagusa, Mersin 10,
Turkey
omar.ramadan@emu.edu.tr

Abstract. Unconditionally stable formulations of the stretched coordinates perfectly matched layer (SCPML) are presented for truncating linear dispersive finite difference time domain (FDTD) grids. In the proposed formulations, the Crank Nicolson and the Bilinear frequency approximation techniques are incorporated with the SCPML to obtain the update equations for the field components in linear dispersive media. Numerical example carried out in one dimensional Lorentz dispersive FDTD domain is included and it has been observed that the proposed formulations not only give accurate results but also remove completely the stability limit of the conventional FDTD algorithm.

1 Introduction

The perfectly matched layer (PML), introduced by Berenger [1], has been shown to be one of the most effective finite difference time domain (FDTD) [2] absorbing boundary conditions (ABCs). This type of ABCs was first introduced for lossless FDTD domains and based on splitting each field component into two subcomponents. Different PML formulations based on the stretched coordinate PML (SCPML) [3], the anisotropic PML (APML) [4], and the nearly PML (NPML) [5] have been introduced for truncating general FDTD domains. Nevertheless, the stability of these formulations is limited by the Courant Friedrichs Lewy (CFL) limit. As the CFL limit is determined by the smallest cell size in the domain, the FDTD analysis of fine geometric structures requires large number of time iterations. Hence, the elimination of the CFL stability limit is one of the latest challenges in the FDTD research.

Recently, unconditionally stable PML formulations based on the alternating direction implicit FDTD (ADI-FDTD) algorithm [6], have been introduced for truncating lossless or lossy [7], and linear dispersive FDTD domains [8]. These formulations, however, are based on splitting each time step into two sub-steps: 1) $n \rightarrow n + 1/2$ and 2) $n + 1/2 \rightarrow n + 1$. In addition, it has been observed that the accuracy of these formulations decreases as the time step increases due to the

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 388–397, 2007.
© Springer-Verlag Berlin Heidelberg 2007

increase of the numerical dispersion error of the ADI-FDTD scheme [9]. Very recently, more accurate and unconditionally stable formulations of the PML, based on incorporating the Crank Nicolson (CN) scheme into the SCPML [3], have been introduced [10]. The main advantage of these formulations is that it allows updating the field components from the n to $n+1$ time step in a single iteration rather that the two sub-iterations needed in the ADI-FDTD method. Nevertheless, these formulations are valid only for modelling non-dispersive FDTD domains.

In this paper, unconditionally stable SCPML formulations are introduced for truncating linear dispersive FDTD domains. The proposed method, named as the CN-SCPML, is based on incorporating Bilinear frequency approximation technique [11] and the CN scheme into the FDTD implementations of the SCPML formulations. The proposed formulations allow updating the field components in a single time iteration rather that the two sub-iterations needed in the ADI-FDTD method. Numerical example carried out in one dimensional (1-D) linear Lorentz dispersive FDTD domain show that the proposed formulations are both accurate and unconditionally stable.

The paper is organized in the following manner. In section 2, the formulations of the proposed scheme are presented. Section 3 and 4 include, respectively, the stability and the numerical dispersion analysis of the proposed formulations. In section 5, the results of a numerical test are included to show the effectiveness of the proposed method, and finally, summary and conclusions are included in section 6.

2 Formulation

Using the SCPML formulation [3], the *normalized* field equations for an x-directed, z-polarized transverse electromagnetic (TEM) wave propagating in 1-D source free homogenous, isotropic, and lossy medium can be written in the frequency domain as

$$j\omega D_z = c\frac{1}{S_x}\frac{\partial}{\partial x}H_y \tag{1}$$

$$j\omega H_y = c\frac{1}{S_x}\frac{\partial}{\partial x}E_z \tag{2}$$

$$D_z = \widehat{\varepsilon}_r(\omega)E_z \tag{3}$$

where c is the speed of light in vacuum, $\widehat{\varepsilon}_r(\omega)$ is the medium relative permittivity, and S_x is the PML stretched coordinate variable defined [3] as

$$\frac{1}{S_x} = \frac{1}{1+\frac{\sigma_x/\varepsilon_o}{j\omega}} = 1 - \frac{\sigma_x/\varepsilon_o}{j\omega + \sigma_x/\varepsilon_o} \tag{4}$$

where σ_x is the conductivity profile along the $x-$direction in the PML region. Using (4) and noting that $j\omega \longrightarrow \partial/\partial t$, (1) and (2) can be written in the time domain as

$$\frac{\partial}{\partial t}D_z = c\frac{\partial}{\partial x}H_y - f_{zx} \tag{5}$$

$$\frac{\partial}{\partial t}H_y = c\frac{\partial}{\partial x}E_z - g_{yx} \tag{6}$$

where f_{zx} and g_{yx} are given by

$$\frac{\partial}{\partial t}f_{zx} + \frac{\sigma_x}{\varepsilon_o}f_{zx} = c\frac{\sigma_x}{\varepsilon_o}\frac{\partial}{\partial x}H_y \tag{7}$$

$$\frac{\partial}{\partial t}g_{yx} + \frac{\sigma_x}{\varepsilon_o}g_{yx} = c\frac{\sigma_x}{\varepsilon_o}\frac{\partial}{\partial x}E_z \tag{8}$$

Using the Crank-Nicolson scheme, (5)-(8) can be discretized as

$$D_{z_i}^{n+1} = D_{z_i}^{n} + \chi\left[H_{y_{i+1/2}}^{n+1} - H_{y_{i-1/2}}^{n+1} + H_{y_{i+1/2}}^{n} - H_{y_{i-1/2}}^{n}\right] - \frac{\Delta t}{2}\left(f_{zx_i}^{n+1} + f_{zx_i}^{n}\right) \tag{9}$$

$$H_{y_{i+1/2}}^{n+1} = H_{y_{i+1/2}}^{n} + \chi\left[E_{z_{i+1}}^{n+1} - E_{z_i}^{n+1} + E_{z_{i+1}}^{n} - E_{z_i}^{n}\right] - \frac{\Delta t}{2}\left(g_{yx_{i+1/2}}^{n+1} + g_{yx_{i+1/2}}^{n}\right) \tag{10}$$

$$f_{zx_i}^{n+1} = r_{0_i}f_{zx_i}^{n} + r_{1_i}\frac{c}{\Delta x}\left[H_{y_{i+1/2}}^{n+1} - H_{y_{i-1/2}}^{n+1} + H_{y_{i+1/2}}^{n} - H_{y_{i-1/2}}^{n}\right] \tag{11}$$

$$g_{yx_{i+1/2}}^{n+1} = r_{0_{i+1/2}}g_{yxi+1/2}^{n} + r_{1_{i+1/2}}\frac{c}{\Delta x}\left[E_{z_{i+1}}^{n+1} - E_{z_i}^{n+1} + E_{z_{i+1}}^{n} - E_{z_i}^{n}\right] \tag{12}$$

where Δx is the space cell size, Δt is the time step, $\chi = c\Delta t/2\Delta x$, and

$$r_{0_i} = \frac{1-q_i}{1+q_i}, \text{ and } r_{1_i} = \frac{q_i}{1+q_i} \tag{13}$$

with $q_i = \Delta t\sigma_{x_i}/2\varepsilon_o$. For a linear dispersive medium $\widehat{\varepsilon}_r(\omega)$, used in (3), can be written in the $s-$domain as

$$\widehat{\varepsilon}_r(s) = \frac{\sum_{m=0}^{L}\widetilde{b}_m s^m}{\sum_{m=0}^{L}\widetilde{a}_m s^m} \tag{14}$$

where $\widetilde{b}_m$ and $\widetilde{a}_m$, $(m = 0, 1, \cdots, L)$, are the coefficients of the rational polynomial and L is the maximum order of the dispersive medium. Using the Bilinear frequency approximations technique [11],

$$s \Rightarrow j\omega \Rightarrow \frac{2}{\Delta t}\frac{1-\mathcal{Z}^{-1}}{1+\mathcal{Z}^{-1}} \tag{15}$$

equation (14) can be written in the $\mathcal{Z}$-domain as

$$\widehat{\varepsilon}_r(\mathcal{Z}) = \frac{\sum_{m=0}^{L} b_m \mathcal{Z}^{(1-m)}}{\sum_{m=0}^{L} a_m \mathcal{Z}^{(1-m)}} \tag{16}$$

where the coefficients b_m and a_m, $(m = 0, 1, \cdots, L)$ are related to $\widetilde{b}_m$ and $\widetilde{a}_m$, and the time step Δt. Taking the $\mathcal{Z}$-transform of (3) and using (15), D_z can be

written in the discrete time domain by using the $\mathcal{Z}$-transform relation $\mathcal{Z}^{-m}G(\mathcal{Z}) \rightarrow G^{n-m}$ [11] as

$$D_{z_i}^{n+1} = \frac{b_0}{a_0} E_{z_i}^{n+1} + \Psi_{z_i}^{n} \tag{17}$$

where

$$\Psi_{z_i}^{n} = \frac{1}{a_0} \sum_{m=1}^{L} \left(b_m E_{z_i}^{n+(1-m)} - a_m D_{z_i}^{n+(1-m)} \right) \tag{18}$$

Substituting (17) into (9), the following can be obtained:

$$\begin{aligned} E_{z_i}^{n+1} = C_0 \left(D_{z_i}^{n} - \Psi_{z_i}^{n} \right) + \chi C_0 \Big[H_{y_{i+1/2}}^{n+1} - H_{y_{i-1/2}}^{n+1} + H_{y_{i+1/2}}^{n} \\ - H_{y_{i-1/2}}^{n} \Big] - \frac{\Delta t}{2} C_0 \left(f_{zx_i}^{n+1} + f_{zx_i}^{n} \right) \end{aligned} \tag{19}$$

where $C_0 = a_0/b_0$. From (19), it can be seen that E_z^{n+1} can not be updated directly as it depends on H_y^{n+1}. Therefore, by substituting (10), (11) and (12) into (19), and after some manipulations, an implicit update for E_z^{n+1} can be obtained from

$$\begin{aligned} &-A_i^{-} E_{z_{i-1}}^{n+1} + A_i E_{zi}^{n+1} - A_i^{+} E_{z_{i+1}}^{n+1} = C_0 D_{z_i}^{n} - C_0 \Psi_{z_i}^{n} - C_0 \frac{\Delta t}{2} \left(1 + r_{0_i}\right) f_{zx_i}^{n} \\ &+ C_0 \left(1 - r_{1_i}\right) \chi \times \Bigg[2 \left(H_{y_{i+1/2}}^{n} - H_{y_{i-1/2}}^{n} \right) - \frac{\Delta t}{2} \left(1 + r_{0_{i+1/2}}\right) g_{yx_{i+1/2}}^{n} \\ &- \frac{\Delta t}{2} \left(1 + r_{0_{i-1/2}}\right) g_{yx_{i-1/2}}^{n} \Bigg] + A_i^{-} E_{z_{i-1}}^{n} - \left(A_i - 1\right) E_{z_i}^{n} + A_i^{+} E_{z_{i+1}}^{n} \end{aligned} \tag{20}$$

where

$$A_i^{\pm} = \Lambda(1 - r_{1_{i\pm1/2}}), \text{ and } A_i = 1 + \Lambda(2 - r_{1_{i+1/2}} - r_{1_{i-1/2}}) \tag{21}$$

with $\Lambda = \chi^2 C_0 (1 - r_{1_i})$. It is clear that the left hand side of (20) forms a tri-diagonal matrix and the corresponding linear system of equations can be solved for E_z^{n+1} easily [6]. Once E_z^{n+1} is obtained, g_{yx}^{n+1}, H_y^{n+1}, and f_{zx}^{n+1} can be updated explicitly from (12), (10) and (11), respectively.

It must be noted that the above formulations are applied in the PML regions at the domain boundaries. In non-PML regions, i.e., in the inner FDTD domain, it is only required to set the coefficients r_0 and r_1 defined in (13) to unity and zero, respectively. This is because the PML conductivity is zero in the inner FDTD domain. Furthermore, it should be noted that the coefficients used in (17) depend on the relative complex permittivity ($\widehat{\varepsilon}_r(\omega)$) of the domain of interest. In this paper, the computational domain was assumed to have second order frequency dispersion, i.e., linear Lorentz type medium. In this case, $\widehat{\varepsilon}_r(\omega)$ can be written as

$$\widehat{\varepsilon}_r(\omega) = \varepsilon_\infty + \frac{\Delta_\varepsilon \omega_0^2}{\omega_0^2 + j2\delta\omega - \omega^2} \tag{22}$$

where ε_s is the static permittivity, ε_∞ is the infinite permittivity, $\Delta_\varepsilon = \varepsilon_s - \varepsilon_\infty$, ω_0 is the resonance radial frequency, and δ is the damping constant. Transforming (22) into the $\mathcal{Z}$-domain by using (15), the coefficients of (17) can be obtained as

$$\begin{aligned} &b_0 = \varepsilon_\infty + 2\varepsilon_\infty\overline{\delta} + \varepsilon_s\overline{\omega}_0^2,\ b_1 = -2(\varepsilon_\infty - \varepsilon_s\overline{\omega}_0^2)\,, \\ &b_2 = \varepsilon_\infty - 2\varepsilon_\infty\overline{\delta} + \varepsilon_s\overline{\omega}_0^2,\ \ a_0 = 1 + 2\overline{\delta} + \overline{\omega}_0^2, \\ &a_1 = -2(1 - \overline{\omega}_0^2)\,,\ \texttt{and}\ \ a_2 = 1 - 2\overline{\delta} + \overline{\omega}_0^2 \end{aligned} \tag{23}$$

with $\overline{\omega}_0 = \omega_0\Delta_t/2$, and $\overline{\delta} = \delta\Delta_t/2$.

3 Stability Analysis

Consider a plane wave propagating in the inner FDTD domain ($\sigma_x = 0$) along the x-direction with field components expressed in the following form:

$$\Psi_i^n = \Psi_0 e^{j\left(\omega n\Delta_t - \widetilde{k}_x i\Delta_x\right)} \tag{24}$$

where $j = \sqrt{-1}$, Ψ_0 is the field component amplitude, and $\widetilde{k}_x$ is the numerical wave-number. Substituting (24) into (9) and (10) and noting that the auxiliary variables f_{zx} and g_{yx} are zero in the inner FDTD domain and then imposing the $\mathcal{Z}$-transform relation $\mathcal{Z}\{G^{n-m}\} = \mathcal{Z}^{-m}G(\mathcal{Z})$, the following can be obtained:

$$(\mathcal{Z} - 1)\, D_0 = -js\,(\mathcal{Z} + 1)\sin\left(\frac{\widetilde{k}_x\Delta_x}{2}\right) H_0 \tag{25}$$

and

$$(\mathcal{Z} - 1)\, H_0 = -js\,(\mathcal{Z} + 1)\sin\left(\frac{\widetilde{k}_x\Delta_x}{2}\right) E_0 \tag{26}$$

where s is the courant number defined as $s = c\Delta_t/\Delta_x$. Equations (25) and (26) can be combined as

$$(\mathcal{Z} - 1)^2\, D_0 + \nu^2\,(\mathcal{Z} + 1)^2\, E_0 = 0 \tag{27}$$

where $\nu^2 = s^2\sin^2\left(\widetilde{k}_x\Delta_x/2\right)$. Using a similar procedure, (17) can be written in the $\mathcal{Z}$-domain for linear Lorentz dispersive media as

$$\left(a_0\mathcal{Z}^2 + a_1\mathcal{Z} + a_2\right) D_0 = \left(b_0\mathcal{Z}^2 + b_1\mathcal{Z} + b_2\right) E_0 \tag{28}$$

Substituting (28) into (27), the following stability polynomial can be obtained:

$$S(Z) = (\mathcal{Z} - 1)^2\left(b_0\mathcal{Z}^2 + b_1\mathcal{Z} + b_2\right) + \nu^2\,(\mathcal{Z} + 1)^2\left(a_0\mathcal{Z}^2 + a_1\mathcal{Z} + a_2\right) \tag{29}$$

To ensure that the proposed scheme is stable, the roots of $S(\mathcal{Z})$ must lie inside or on the unit circle of the $\mathcal{Z}$-plane. Alternatively, the stability of the proposed

scheme can be studied by using the Routh-Hurwitz criterion [12] by transforming (29) into the r-plane by using the relation

$$\mathcal{Z} = \frac{r+1}{r-1} \tag{30}$$

Hence, (29) can be written in the r-plane as

$$S(r) = \overline{\omega}_0^2\nu^2 r^4 + 2\overline{\delta}\nu^2 r^3 + (\nu^2 + \varepsilon_s\overline{\omega}_0^2)r^2 + 2\varepsilon_\infty\overline{\delta}r + \varepsilon_\infty \tag{31}$$

In this transformation, the exterior of the unit circle in the $\mathcal{Z}$-plane is mapped onto the right half of the r-plane. Therefore, if $S(r)$ has no roots in the right half of the r-plane, then $S(\mathcal{Z})$ will not have any root outside the unit circle of the $\mathcal{Z}$-plane. It must be noted that $S(r)$ polynomial will not have roots in the right half of the r-plane if all entries of the first column of the Routh table, which can be constructed from $S(r)$, are non-negative [12]. To this end, the Routh table can be constructed from (31) [12] as

$\overline{\omega}_0^2\nu^2$	$\nu^2 + \varepsilon_s\overline{\omega}_0^2$	ε_∞
$2\overline{\delta}\nu^2$	$2\overline{\delta}\varepsilon_\infty$	0
$\nu^2 + (\varepsilon_s - \varepsilon_\infty)\,\overline{\omega}_0^2$	ε_∞	0
$\frac{2\overline{\delta}\varepsilon_\infty(\varepsilon_s-\varepsilon_\infty)\overline{\omega}_0^2}{\nu^2+(\varepsilon_s-\varepsilon_\infty)\overline{\omega}_0^2}$	0	0
ε_∞	0	0

(32)

Forcing all entries of the first column of the Routh table to be non-negative quantities, the following conditions can be obtained:

$$\begin{aligned} \delta &\geq 0 \\ \varepsilon_s &\geq \varepsilon_\infty \end{aligned} \tag{33}$$

It is clear from (33) that there is no restriction imposed on the time step Δ_t for the stability of the proposed formulations, hence, the formulations introduced in this paper are always stable.

4 Numerical Dispersion Analysis

The numerical dispersion of the proposed scheme can be easily found by evaluating the stability polynomial on the unit circle of the $\mathcal{Z}$-plane, i.e., by letting $\mathcal{Z} = e^{j\omega\Delta_t}$ in (29) [12]. By doing this and after some manipulations, the following can be obtained:

$$s^2 \sin^2\left(\frac{\widetilde{k}_x\Delta_x}{2}\right) = \widetilde{\epsilon}\tan^2\left(\frac{\omega\Delta_t}{2}\right) \tag{34}$$

where $\widetilde{\epsilon}$ is the numerical permittivity defined as

$$\begin{aligned} \widetilde{\epsilon} &= \frac{\varepsilon_s\overline{\omega}_0^2 + j2\overline{\delta}\varepsilon_\infty\tan(\frac{\omega\Delta_t}{2}) - \varepsilon_\infty\tan^2(\frac{\omega\Delta_t}{2})}{\overline{\omega}_0^2 + j2\overline{\delta}\tan(\frac{\omega\Delta_t}{2}) - \tan^2(\frac{\omega\Delta_t}{2})} \\ &= \varepsilon_\infty + \frac{(\varepsilon_s - \varepsilon_\infty)\,\overline{\omega}_0^2}{\overline{\omega}_0^2 + j2\overline{\delta}\tan(\frac{\omega\Delta_t}{2}) - \tan^2(\frac{\omega\Delta_t}{2})} \end{aligned} \tag{35}$$

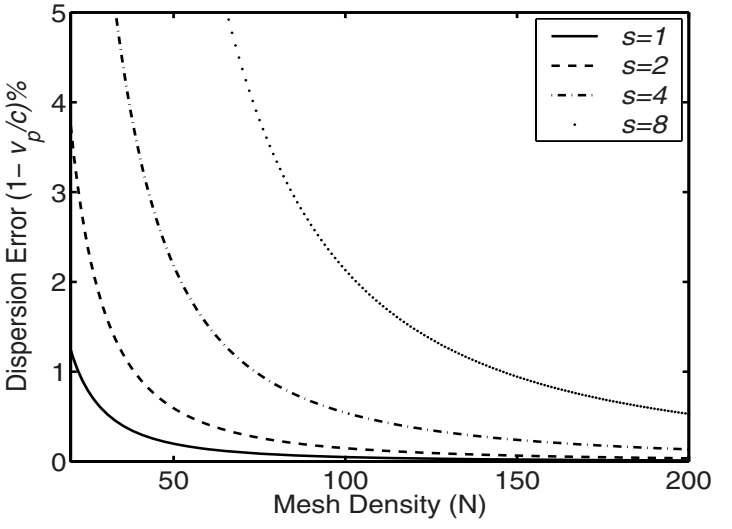


Fig. 1. Numerical dispersion error as a function of the mesh density N for different values of the courant number

Solving for $\widetilde{k}_x$ from (34), the following can be obtained:

$$\widetilde{k}_x = \frac{2}{\Delta_x} \sin^{-1}\left[\frac{1}{s}\tan\left(\frac{\omega\Delta_t}{2}\right)\sqrt{\widetilde{\epsilon}}\right] \tag{36}$$

Using (36), the numerical wavelength is

$$\widetilde{\lambda} = \frac{2\pi}{Re(\widetilde{k}_x)} \tag{37}$$

Hence, the numerical phase velocity v_p can be computed as

$$v_p = \frac{2\pi f}{Re(\widetilde{k}_x)} = \frac{\pi}{N}\frac{c}{Re\left[\sin^{-1}\left(\frac{1}{s}\tan\left(\frac{\pi s}{N}\right)\sqrt{\widetilde{\epsilon}}\right)\right]} \tag{38}$$

where N is the mesh density defined as the number of cells per wavelength, i.e., $N = \lambda/\Delta_x$. Fig. 1 shows the relative error in the numerical wave velocity $(p = 1 - v_p/c)$ as a function of N for different values of s. In this test, the space cell size was taken as $1 \times 10^{-12}m$ and the Lorentz domain dielectric parameters are chosen similar to those used in [13]. As can be seen from Fig. 1, the error decreases as N increases and also increases as s increases. At $s = 1$ and for $N = 200$, the wave velocity receives approximately 0.01% error. When s is increased to 4, the error will be 0.13% at the same mesh density. Similarly, when s is further increased to 8, the error will be increased to 0.53%.

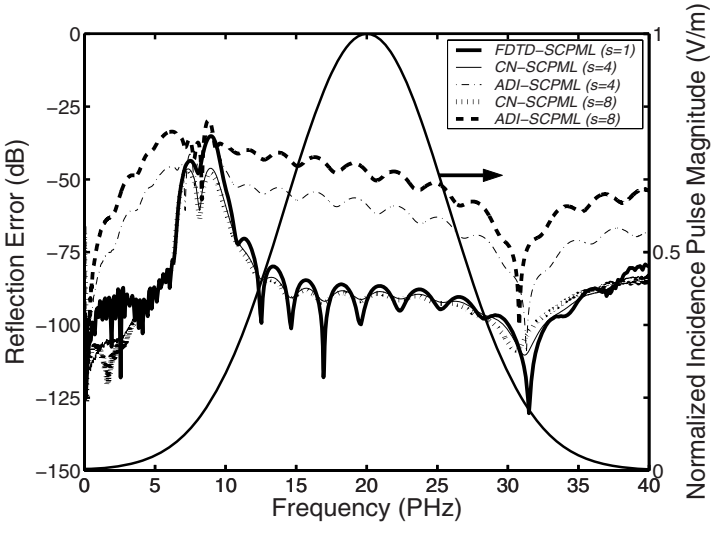


Fig. 2. Reflection coefficient spectrum for Lorentz medium as obtained with the FDTD-SCPML, ADI-SCPML and CN-SCPML formulations

5 Simulation Study

To validate the proposed formulations, a $z-$polarized Gaussian pulse was excited at the center of 1-D linear Lorentz medium with the parameters of $\varepsilon_\infty = 1$, $\varepsilon_s = 2.25$, $\omega_0 = 4\times10^{16} rad/s$, and $\delta = 0.28\times10^{16} s^{-1}$ [13]. The computational domain extends in the $x-$direction and assumed to be $100\Delta_x$ where $\Delta_x = 1\times10^{-10} m$. Both ends of the computational domain were terminated by eight additional PML layers with the parameters of PML[8, 2, 0.001%], as defined in [1]. The simulation was carried out for the first 32768 time steps.

To measure the PML reflection error, a reference solution is needed. In this study, a larger domain with the size of $1000\Delta_x$ and truncated by 32 additional PML layers with the parameters of PML[32, 4, 0.001%] was used. Figure 2 shows the frequency spectrum of the reflection coefficient of the proposed CN-SCPML along with the excitation source used in this test. The reflection coefficient was computed one cell away from the PML/computational domain interface as

$$\mathcal{R}_{dB}(f) = 20\log_{10}\left|\frac{\mathcal{F}\left\{E_z^R(t) - E_z^T(t)\right\}}{\mathcal{F}\left\{E_z^R(t)\right\}}\right| \tag{39}$$

where $\mathcal{F}\{.\}$ is the Fourier transform operation, $E_z^T(t)$ is the field computed using the test domain, and $E_z^R(t)$ is the reference field computed using the larger domain. For the purpose of comparison, the results obtained using the conventional FDTD implementations of the SCPML (FDTD-SCPML) [3] and for the ADI-FDTD implementations of the SCPML (ADI-SCPML) [8] were also shown in Fig. 2. In this study, it must be noted that the reference field was computed

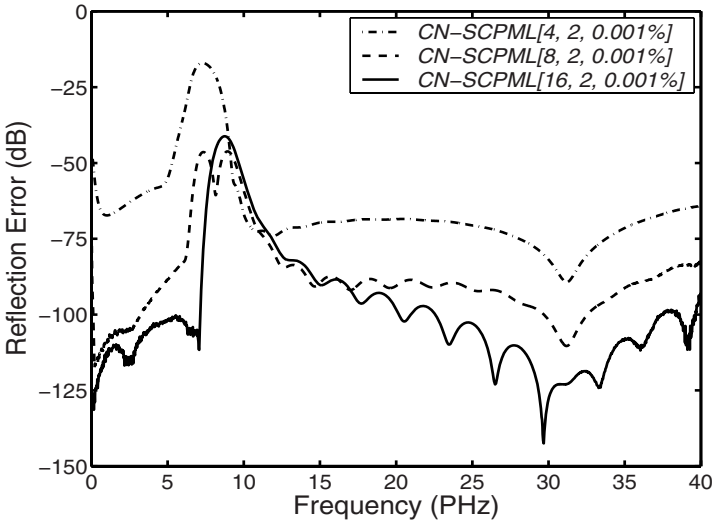


Fig. 3. Reflection coefficient spectrum for Lorentz medium as obtained with the proposed CN-SCPML formulations for different PML layers

for each value of the courant number in order to isolate PML reflection error from the numerical dispersion error of the discrete FDTD schemes [7]. As can be seen form Fig. 2, it is apparent that the proposed CN-SCPML formulations do not only remain stable for $s > 1$ but also are in a good agreement with the conventional FDTD-SCPML. On the other hand, the accuracy of the ADI-SCPML formulations decreases as the value of the courant number increases. It must also be pointed out that there is a sharp increase in the reflection coefficient near the resonance frequency $f_0 = 6.3662PHz$. However, this is not important because this frequency range is strongly attenuated for the chosen dielectric parameters. The performance of the proposed formulations was also measured as a function of the number of PML layers. Fig. 3 shows the reflection error of the CN-SCPML for different PML layers and with a courant number $s = 4$. As expected, the reflection errors decreases as the number of NPML layers increases.

6 Conclusion

In this paper, unconditionally stable formulations of the SCPML are presented for truncating linear dispersive FDTD domains. In the proposed formulations, the CN and the Bilinear frequency approximation techniques are used to obtain the update equations of the field components in the dispersive medium. It has been shown through numerical example carried out in 1-D Lorentz dispersive FDTD domain that the proposed CN-SCPML scheme is unconditionally stable and maintains the accuracy of the conventional FDTD-SCPML formulations. Hence, the simulation time can be decreased by increasing the time step.

References

1. Berenger, J.-P.: A perfectly matched layer for the absorption of electromagnetic waves. Journal of Computational Physics 114, 185–200 (1994)
2. Taflove, A., Hangess, S.C.: Computational electrodynamics: The Finite-Difference Time-Domain Method, 2nd edn. Artech-House, Boston (2000)
3. Chew, W.C., Weedon, W.H.: A 3-D perfectly matched medium from modified Maxwell's equation with stretched coordinates. Microwave and Optical Technology Letters 7(13), 599–604 (1994)
4. Gedney, S.D.: An anisotropic perfectly-matched layer-absorbing medium for the truncation of FD-TD lattices. IEEE Transactions on Antennas and Propagation 44(12), 1630–1639 (1996)
5. Cummer, S.A.: A simple, nearly perfectly matched layer for general electromagentic media. IEEE Microwave and Wireless Components Letters 13(3), 128–130 (2003)
6. Namiki, T.: A new FDTD algorithm based on alternating-direction implicit method. IEEE Transactions on Microwave Theory and Techniques 47(10), 2003–2007 (1999)
7. Gedney, S.D., Liu, G., Roden, J.A., Zhu, A.: Perfectly matched layer media with CFS for an unconditionally stable ADI-FDTD method. IEEE Transactions on Antennas and Propagation 49(11), 1554–1559 (2001)
8. Ramadan, O.: Unconditionally stable ADI-FDTD implementation of PML for frequency dispersive debye media. Electron. Lett. 40(4), 230–232 (2004)
9. Carcia, S.G., Lee, T.W., Hagness, S.C.: On the accuracy of the ADI-FDTD method. IEEE Antennas and Wireless Propagation Letters 1, 31–34 (2002)
10. Ramadan, O.: Generalized unconditionally stable Crank-Nicolson PML formulations for truncating FDTD domains. In: The 35^{th} European Microwave Conference (EuMC), Paris, France (October 4-6, 2005)
11. Proakis, J.G., Manolakis, D.G.: Digital signal processing: principles, algorithms and applications, 3rd edn. Prentice-Hall, Englewood Cliffs (1995)
12. Pereda, J.A., Vielva, L.A., Vegas, A., Prieto, A.: Analyzing the stability of the FDTD technique by combining the Von Neumann method with Routh-Hurwitz criterion. IEEE Transactions on Microwave Theory and Techniques 49(2), 377–381 (2001)
13. Joseph, R.M., Hagness, S.C., Taflove, A.: Direct time integration of Maxwell's equations in linear dispersive media with absorption for scattering and propagation of femtosecond electromagnetic pulses. Optics Letters 16(18), 1412–1414 (1991)

A Novel Congestion Control Scheme for Elastic Flows in Network-on-Chip Based on Sum-Rate Optimization

Mohammad S. Talebi[1], Fahimeh Jafari[1,3], Ahmad Khonsari[2,1], and Mohammad H. Yaghmae[3]

[1] IPM, School of Computer, Tehran, Iran.,
[2] ECE Department, University of Tehran
[3] Ferdowsi University of Mashhad,
mstalebi@ipm.ir, jafari@ipm.ir, ak@ipm.ir, hyaghmae@ferdowsi.um.ac.ir

Abstract. Network-on-Chip (NoC) has been proposed as an attractive alternative to traditional dedicated busses in order to achieve modularity and high performance in the future System-on-Chip (SoC) designs. Recently, end-to-end congestion control has gained popularity in the design process of network-on-chip based SoCs. This paper addresses a congestion control scenario under traffic mixture which is comprised of Best Effort (BE) traffic or elastic flow and Guaranteed Service (GS) traffic or inelastic flow. We model the desired BE source rates as the solution to a rate-sum maximization problem which is constrained with link capacities while preserving GS traffic services requirements at the desired level. We proposed an iterative algorithm as the solution to the maximization problem which has the advantage of low complexity and fast convergence. The proposed algorithm may be implemented by a centralized controller with low computation and communication overhead.

1 Introduction

The *Systems-on-Chip* (SoC) was first designed as a tightly interconnected set of cores, where all components share the same system clock, and the communication between components is via shared-medium busses. With the advance of the semiconductor technology, the enormous number of transistors available on a single chip allows designers to integrate dozens of IP blocks together with large amounts of embedded memory. Such IPs consist of CPU or DSP cores, video stream processors, high-bandwidth I/O, routers, *etc*. As more and more cores are integrated into a single chip, it is becoming increasingly difficult to meet the design constraints while still using the old design methodologies for SoC designs. Shared-medium busses do not scale well, and do not fully utilize potentially available bandwidth. As the features sizes shrink, and the overall chip size relatively increases, interconnects start behaving as lossy transmission lines. Crosstalk, electromagnetic interference, and switching noise cause higher incidence of data errors. Line delays have become very long as compared to gate delays causing synchronization problems between cores. A significant amount of power is dissipated on long interconnects and in clocking network. This trend only

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, pp. 398 – 409, 2007.
© Springer-Verlag Berlin Heidelberg 2007

worsens as the clock frequencies increase and the features sizes decrease. Lowering the power supplies and designing smaller logic swing circuits decreases the overall power consumption at the cost of higher data errors.

One solution to these problems is to treat SoCs implemented using *micro-networks*, or *Networks on Chips* (NoCs). Networks have a much higher bandwidth due to multiple concurrent connections. They have regular structure, so the design of global wires can be fully optimized and as a result, their properties are more predictable. Regularity enables design modularity, which in turn provides a standard interface for easier component reuse and better interoperability. Overall performance and scalability increase since the networking resources are shared.

Networking model decouples the communication layers so that design and synthesis of each layer is simpler and can be done separately. In addition, decoupling enables easier management of power consumption and performance at the level of communicating cores. Generally, the concept of NoC which was introduced in [1][2], suggests that different modules would be connected by a simple network of shared links and routers. Examples of NoCs are Æthereal [3], Mango [4] and Xpipes [5]. NoCs provide communication services to IPs. Communication services with guarantees on throughput and latency enable predictable system design. Guarantees are given by reserving communication resources in the NoC (e.g. wires and buffers). Although necessary for hard real-time applications, this results in poor resource utilization for applications that require Variable-Bit Rate (VBR) communication. According to the nature of such kind of traffic, this is also called *Inelastic Flow*. Best Effort service (BE) is another kind of communication services which doesn't need any guarantees on latency and bandwidth. According to the nature of this kind of traffic, this is also called *Elastic Flow*. Elastic Flow or BE service can give high resource utilization by using unreserved or unused resources. However, BE traffic is prone to network congestion. Æthereal [3] and Mango [4] are examples of NoCs that provide both GS and BE services.

Networks with BE services should have a strategy to avoid congestion. The congestion control in NoCs is a novel problem for the resource constrained on-chip designs. During the past two decades, many strategies for congestion control have been proposed for off-chip networks [6, 7, 8]. Congestion control for on-chip networks is still a novel issue, however this problem has been investigated by several researchers [9]-[11]. In [9], a prediction-based flow control strategy for on-chip networks has been proposed where each router predicts future buffer fillings to detect future congestion problems. The buffer filling predictions are based on a router model. Dyad [10] solves congestion problem by switching from deterministic to adaptive routing when the NoC gets congested. In [11] the link utilization has been used as congestion measure and the controller determines the appropriate loads for the BE sources. All of the aforementioned work has dealt with this issue using the predictive control approach to overcome the congestion in the network. As the NoC architecture is similar to a regular data network, in this paper we have used an optimization approach over Best Effort source rates to control the flow.

The main purpose of this paper is to present a congestion control as the solution to a sum-rate maximization problem for choosing the rate of BE sources. We present an algorithm as the solution to the optimization problem and prove its convergence. To evaluate the performance of the proposed approach, we simulate the congestion

control algorithm under a NoC-based scenario. Similar to [10], we have used a controller to implement the proposed algorithm; however our approach is completely different from [10].

This paper is organized as follows. In section 2 we present the system model and formulate the underlying optimization problem for BE flow control. In section 3 we solve the optimization problem using an iterative algorithm and propose the solution as a centralized congestion control algorithm to be implemented as a controller. In section 4 we analyze the convergence behavior of the proposed algorithm and prove the underlying theorem of its convergence. In section 5 we present the simulation results. Finally, the section 6 concludes the paper and states some future work directions.

2 System Model

We consider a NoC with two dimensional mesh topology and wormhole routing. In wormhole networks, each packet is divided into a sequence of *flit*s which are transmitted over physical links one by one in a pipeline fashion. The NoC architecture is assumed to be lossless, and packets traverse the network on a shortest path using a deadlock free XY routing.

We model the congestion control problem in NoC as the solution to an optimization problem. For more convenience, we turn the aforementioned NoC architecture into a mathematical model as in [8]. In this respect, we consider NoC as a network with a set of bidirectional links L and a set of sources S. A source consists of Processing Elements (PEs), routers and Input/Output ports. Each link $l \in L$ is a set of wires, busses and channels that are responsible for connecting different parts of the NoC and has a fixed capacity of c_l packets/sec. We denote the set of sources that share link l by $S(l)$. Similarly, the set of links that source s passes through, is denoted by $L(s)$. By definition, $l \in S(l)$ if and only if $s \in L(s)$ [8].

As previously stated, there are two types of traffic in a NoC: Guaranteed Service traffic (GS) or inelastic flow and Best Effort (BE) traffic or elastic flow. For notational convenience, we divide S into two parts, each one representing sources with the same traffic. In this respect, we denote the set of sources with BE and GS traffic by S_{BE} and S_{GS}, respectively. Each link l is shared between the two aforementioned traffics. GS sources will obtain the required amount of the capacity of links and the remainder should be allocated to BE sources.

Our objective is to choose source rates (PE loads) of BE traffics so that to maximize the sum of rates of all BE traffics. Hence the maximization problem can be formulated as:

$$\max_{x_s} \sum_{s \in S_{BE}} x_s \tag{1}$$

subject to:

$$\sum_{s \in S_{BE}(l)} x_s + \sum_{s \in S_{GS}(l)} x_s \le c_l \qquad \forall l \in L \tag{2}$$

$$x_s > 0 \qquad \forall s \in S_{BE} \tag{3}$$

where source rates, i.e. x_s, $s \in S$, are optimization variables.

The constraint Eq. (2) says the aggregate BE source rates passing thorough link l cannot exceed its free capacity, i.e. the portion of the link capacity which has not been allocated to GS sources. The abovementioned problem is in fact *constrained sum-rate maximization.* Such a problem, in general belongs to the class of *Utility-Maximization Problems* for which the utility function of all sources is considered to be *Identity Function*, i.e. $U_s(x_s) = x_s$. Although the general form of Eq. (1) with a general utility function has been investigated by the authors in another work [12], the approach of this paper to solve problem Eq. (1) is completely different from the previous work. In this paper we focus on the primal problem while in [12] the problem is solved via its dual function. For notational convenience, we define:

$$\hat{c}_l = c_l - \sum_{s \in S_{GS}(l)} x_s \tag{4}$$

Hence, Eq. (2) can be rewritten as:

$$\sum_{s \in S_{BE}(l)} x_s \leq \hat{c}_l \qquad \forall l \in L \tag{5}$$

Although problem Eq. (1) is separable across sources, its constraints will remain coupled across the network. Due to coupled nature of such constrained problems, they have to be solved using centralized methods like interior point methods [13]-[15]. Such computations may pose a great overhead on the system. Instead of such methods, we seek to obtain the solution with simpler operations. One way is to use the subgradient method for constrained optimization problems [16] which will be briefly reviewed in the next section.

For notational convenience in solving the problem, we use matrix notation. In this respect, we define Routing matrix, i.e. $R = [R_{ls}]_{L \times S}$, as following:

$$R_{ls} = \begin{cases} 1 & \text{if } s \in S_{BE}(l) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

We also define the source rate vector (for BE traffic) and link capacity vector as $x = (x_s,\ s \in S_{BE})$ and $\hat{c} = (\hat{c}_l,\ l \in L)$, respectively. Therefore problem Eq. (1) can be rewritten in the matrix form as follows:

$$\max_x \ \mathbf{1}^T x \tag{7}$$

subject to:

$$Rx \leq \hat{c} \tag{8}$$

$$x_s > 0 \qquad \forall s \in S_{BE} \tag{9}$$

where $\mathbf{1}$ is a vector with all one.

3 Congestion Control Algorithm

In this section, we will solve the sum-rate optimization problem Eq. (7) using subgradient method for constrained optimization problems [13][16] and present a flow control scheme for BE traffic – or elastic flows - in NoC systems to overcome the congestion. Convergence analysis of the algorithm is to be discussed in the next section.

The subgradient method for constrained optimization problems is very similar to *Poljak's Method* [17]. In this method, for a maximization problem like Eq. (1), the optimization variable vector will be adjusted in the direction to the gradient of the objective function. We briefly review this method in lemma 1 as follows.

Lemma 1. *Consider the constrained maximization problem,*

$$\max_{x} f(x) \tag{10}$$

subject to:

$$f_i(x) \geq 0, \quad i = 1..M \tag{11}$$

with the maximal x^* *and the sequence* $x(t)$ *as*

$$x(t+1) = x(t) + \gamma(t)u(x(t)) \tag{12}$$

where

$$u(x(t)) = \begin{cases} \nabla f(x(t)) & \text{if } x(t) \text{ satisfies (11)} \\ \nabla f_j(x(t)) & \exists j \text{ s.t } \ f_j(x(t)) < 0 \end{cases} \tag{13}$$

where $\gamma(t)$ *is a diminishing step-size rule [13]-[15]. If the* l_2 *-norm of the* $u(x(t))$ *is bounded (Lipschitz Continuity), i.e. there exist* G *such that*

$$\|u\|_2 \leq G \tag{14}$$

and the Euclidian distance of the initial point to the optimal point is bounded, i.e.

$$\left\|x(1) - x^*\right\| \leq D \tag{15}$$

then the sequence $\left\{x(t)\right\}_{t=1}^{\infty}$, *as* $t \to \infty$ *will converge to* x^*.

Proof: See [16].

In the sequel, we will solve the optimization problem Eq. (7) using subgradient method for constrained optimization problems as stated in Lemma 1. Regarding Eq. (12), we should calculate $u(x(t))$. According to Eq. (13), if $x(t)$ is feasible, i.e. $Rx \leq \hat{c}$, we have:

$$u = \nabla \mathbf{1}^T x = \nabla x^T \mathbf{1} = \mathbf{1} \tag{16}$$

otherwise at least one of the constraints should be violated. Assuming the corresponding constraint for link $l^{'}$ is violated, i.e. $\sum_{s\in S_{BE}(l^{'})} x_s > \hat{c}_{l^{'}}$. We can represent this constraint in matrix form as:

$$f_{l^{'}}(x) = \mathbf{e}_{l^{'}}^T(\hat{c} - Rx) < \mathbf{0} \tag{17}$$

where $\mathbf{e}_{l^{'}}$ is the $l^{'}$th unit vector of $\mathbb{R}^L$ space which is zero in all entries except the $l^{'}$th at which it is 1. Therefore, u is given by:

$$u = -\nabla \mathbf{e}_{l^{'}}^T(Rx - \hat{c}) = -R^T\mathbf{e}_{l^{'}} \tag{18}$$

Using Eq. (16) and Eq. (18), the update equation to solve problem Eq. (7) is given by:

$$x(t+1) = x(t) + \gamma(t)u(x(t)) \tag{19}$$

where $u(x(t))$ is given by:

$$u(t) = \begin{cases} \mathbf{1} & \sum_{s\in S_{BE}(l)} x_s(t) \le \hat{c}_l, \forall l \\ -R^T\mathbf{e}_{l^{'}} & \sum_{s\in S_{BE}(l^{'})} x_s(t) > \hat{c}_{l^{'}}, \exists l^{'} \end{cases} \tag{20}$$

Stepsize has an important role on the convergence behavior of the update equation. There are several choices for stepsize, each one belonging to a predefined category and having certain advantages and drawbacks (see [14] and references herein).

In the family of gradient algorithms, for distributed scenarios stepsize is usually chosen to be a small enough constant so that to guarantee the convergence of the algorithm. Constant stepsize is robust in the sense of convergence in time-varying conditions and asynchronous schemes[1]. However, it mainly suffers from slow convergence rate. On the contrary, time-varying stepsizes are defined in such a way to adapt to the error with the desired point, i.e. optimal point of the optimization problem, and hence benefit from much more faster convergence. However, they should be constrained to guarantee that the iterative algorithm will converge.

In our scheme, the algorithm is to be centralized in implementation, and thus we use a time-varying stepsize to take advantage of fast convergence. To this end, we choose $\gamma(t)$ as a time-varying stepsize, to be *square-summable but not summable* [14][16]. In this respect, $\gamma(t)$ satisfies

$$\gamma(t) \ge 0 \qquad \forall t \tag{21}$$

$$\sum_{k=1}^{\infty} \gamma^2(t) < \infty \tag{22}$$

[1] Note that Eq. (19) proposes a synchronous scheme, and may diverge in asynchronous ones, e.g. real world conditions with large delays, etc.

$$\sum_{k=1}^{\infty} \gamma(t) = \infty \tag{23}$$

One typical example that satisfies Eq. (21)-(23), is $\gamma(t) = a/(b+t)$, where $a > 0$ and $b \geq 0$, which we have used in this paper.

Eq. (19) and Eq. (20) together propose an iterative algorithm as the solution to problem Eq. (7). In this respect, optimal source rates for BE sources can be found while satisfying capacity constraints and preserving GS traffic requirements. Thus, the aforementioned algorithm can be employed to control the congestion of the BE traffic in the NoC. The above iterative algorithm is decentralized in the nature and can be addressed in distributed scenarios. However, due to well-formed structure of the NoC, we focus on a centralized scheme; a simple controller can be mounted in the NoC to implement this algorithm. The necessary requirement of such a controller is the ability to accommodate simple mathematical operations as in Eq. (19) and Eq. (20) and the allocation of few wires to communicate congestion control information to nodes with a light GS load. Algorithmic realization of the proposed Congestion-Controller for BE traffic is listed as Algorithm 1.

Algorithm 1. Congestion Control for BE Traffics in NoC

Initialization:

1. **Initialize $\hat{c}_l$ of all links.**
2. **Set source rate vector to zero.**

Loop:

Do until $(\max |x_s(t+1) - x_s(t)| < Error)$

1. $\forall s \in S$ **: Compute new source rate:**

$$x(t+1) = x(t) + \gamma(t) u(x(t))$$

where $\gamma(t)$ **can be selected as** $\gamma(t) = a/(b+t)$ and

$$u(t) = \begin{cases} \mathbf{1} & \sum_{s \in S_{BE}(l)} x_s(t) \leq \hat{c}_l, \ \forall l \\ -R^T \mathbf{e}_{l'} & \sum_{s \in S_{BE}(l')} x_s(t) > \hat{c}_{l'} \end{cases}$$

Output:

Communicate BE source rates to the corresponding nodes.

4 Convergence Analysis

In this section, we investigate the convergence analysis of the proposed algorithm using a time-varying stepsize in Eq. (19). As stated in the previous section, in this paper the stepsize is selected to be *square-summable but not summable* [16].

Theorem 1. *The iterative congestion control scheme proposed by Eq. (19) and Eq. (20) with a time-varying stepsize which satisfies Eq. (21)-(23), will converge to the optimal point of problem Eq. (1).*

***Proof*:** By lemma 1, it is clear that if its assumptions hold, the proof of Theorem is done. First, $u(x(t))$ should admit an upper bound in l_2 -norm. In doing so, it suffices to show that its gradient is upper bounded in l_2 -norm. Considering Eq. (16) and (18), we have

$$\begin{aligned} \|u\|_2 &\le \max\{\|\mathbf{1}\|_2, \left\|-R^T \mathbf{e}_l\right\|_2\} \\ &= S \end{aligned} \tag{24}$$

hence u in l_2 -norm is bounded with at least S .

In the next step, we show that the Euclidian distance of the initial point to the optimal point is bounded at least with D , i.e.

$$\exists D > 0 \qquad \text{s.t.} \qquad \left\|x(1) - x^*\right\|_2 \le D$$

According to Eq. (3), we have $x_s > 0, \ \forall s \in S_{BE}$. On the other hand, optimal source rates are bounded at most with maximum value of link capacities, i.e.

$$\max_s x_s^* \le \max_l \hat{c}_l \le \max_l c_l \tag{25}$$

therefore,

$$\begin{aligned} \left\|x(1) - x^*\right\|_2 &\le \left\|\max x^* - \min x(1)\right\|_2 \\ &= \left\|\max_l c_l - 0\right\|_2 \\ &= L c_{l\max} \end{aligned} \tag{26}$$

and hence the initial Euclidian distance is bounded and Eq. (26) with Eq. (24) completes the proof.

5 Simulation Results

In this section we examine the proposed congestion control algorithm, listed above as Algorithm 1, for a typical NoC architecture. We have simulated a NoC with 4×4 Mesh topology which consists of 16 nodes communicating using 24 shared bidirectional links; each one has a fixed capacity of 1 Gbps. In our scenario, packets traverse the network on a shortest path using a deadlock free XY routing. We also assume that each packet consists of 500 flits and each flit is 16 bit long.

In order to simulate our scheme, some nodes are considered to have a Guaranteed Service data (such as Multimedia, etc.) to be sent to a destination while other nodes,

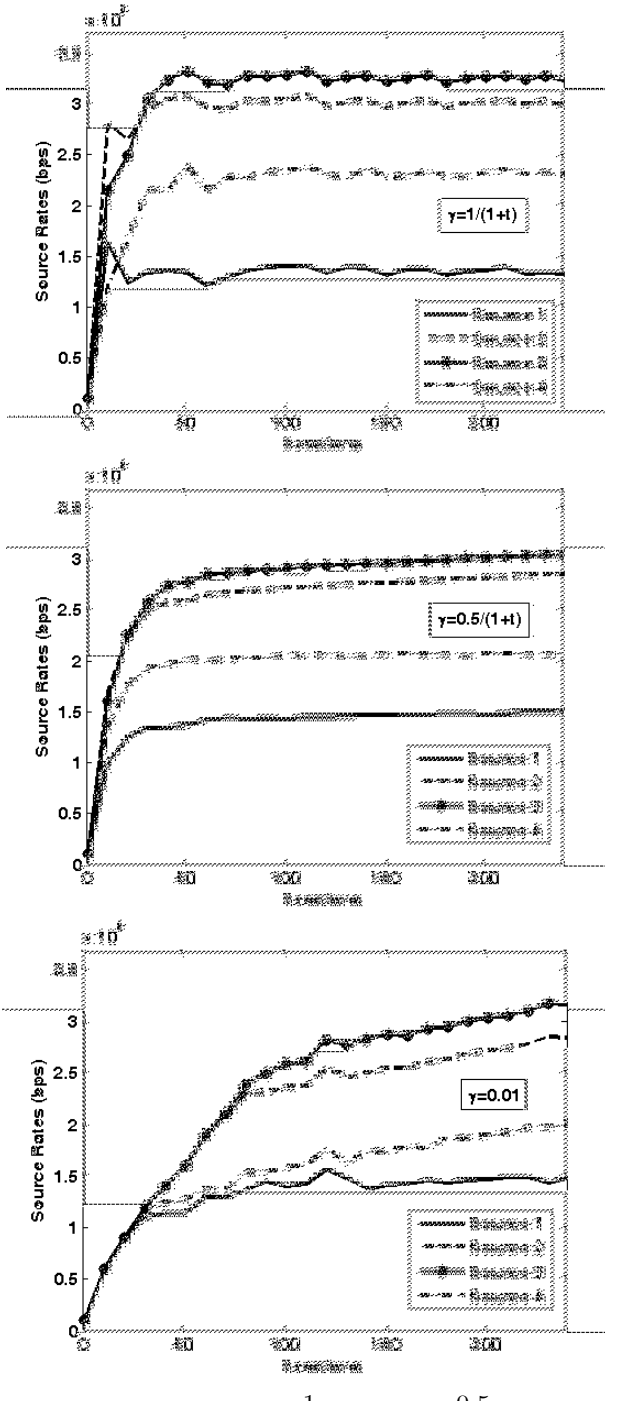


Fig. 1. Source rates for (a) $\gamma = \dfrac{1}{1+t}$, (b) $\gamma = \dfrac{0.5}{1+t}$ and (c) $\gamma = 0.01$

which maybe in the set of nodes with GS traffic, have a Best Effort traffic to be sent. As stated in section 2, GS sources will obtain the required amount of the capacity of links and the remainder should be allocated to BE traffics.

One of the most significant issues of our interest is the convergence behavior of the source rates. We used three different scenarios for step-size; two of them are chosen to be *square-summable but not summable* and the third is set to be constant. For the first two cases, stepsizes are chosen as $\gamma = 1/(1+t)$ and $\gamma = 0.5/(1+t)$ which satisfy Eq. (21)-(23). For the constant case, stepsize is set to be $\gamma = 0.01$. The first and second cases will be comparable with the constant stepsize after about 99 and 49 iterations, respectively.

Variation of source rates for some nodes using aforementioned stepsizes are shown in Fig. 1(a)-(c). Regarding Fig. 1(a), it's apparent that after about 50 iterations, all source rates will be in the vicinity of the steady state point of the algorithm. However, for the second case, Fig. 1(b) reveals that at least 80 iterations needed to have source rates in the vicinity of the optimal point. For the third case, the rate of convergence is even less and at least 150 iterations are needed to fall within the neighborhood of the steady state point of the algorithm. It is clear that compared to the *square-summable but not summable* stepsizes, constant stepsize has much slower rate of convergence. Comparing Fig. 1(a) and 1(b), we realize that the initial value of the stepsize, directly influences the rate of convergence.

In order to have a better insight about the algorithm behavior, the relative error with respect to optimal rates which averaged over all sources, is also shown in Fig. 2. Optimal source rates are obtained using CVX [18] which is MATLAB-based software for solving disciplined convex optimization problems. This figure reveals that *square-summable but not summable* stepsizes can lead to lower relative error in

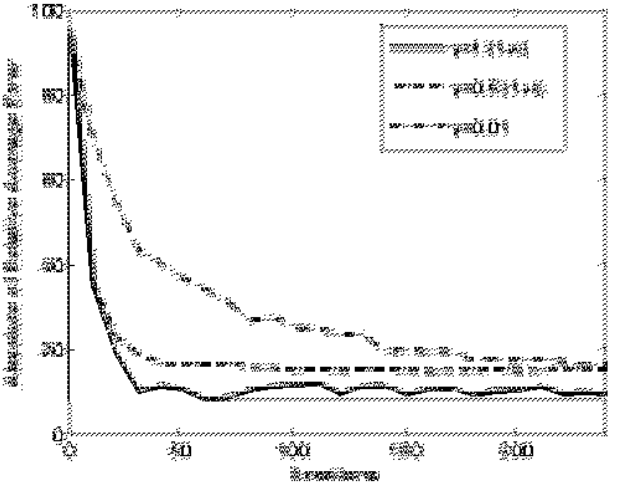


Fig. 2. Average of relative error with respect to optimal solution for the three cases

average with regard to constant stepsize. The faster rate of convergence of the first two cases than the third, can also be verified and it is apparent that the first case slightly acts better from the second in terms of averaged relative error.

6 Conclusion and Future Work

In this paper we addressed the problem of congestion control for BE traffic in NoC systems. Congestion control was modeled as the solution to the sum-rate maximization problem which was solved using subgradient method for constrained optimization problems. This was led to an iterative algorithm which determine optimal BE source rates. We have also studied the realization of the algorithm as a centralized congestion controller and presented a theorem to prove the convergence of the proposed congestion control scheme. Simulation results confirm that the proposed algorithm converges very fast and the computational overhead of the congestion control algorithm is small. Fast convergence of the algorithm also justifies that the delay incurred by the algorithm is very small. Further investigation about the effect of other utility functions on the BE rates and fairness provision is the main direction of our future studies.

References

1. Guerrier, P., Greiner, A.: A Generic Architecture for On-Chip Packet-Switched Interconnections. In: Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE) (2000)
2. Dally, W.J., Towles, B.: Route Packets, Not Wires: On-Chip Interconnection Networks. In: Proc. DAC 2001 (2001)
3. Goossens, K., et al.: The Æthereal network on chip: Concepts, architectures, and implementations. IEEE Design and Test of Computers 22(5) (2005)
4. Bjerregaard, T., et al.: A router architecture for connection oriented service guarantees in the MANGO clockless Network-on-Chip. In: Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE) (2005)
5. Bertozzi, D., et al.: Xpipes: A network-on-chip architecture for gigascale systems-on-chip. IEEE Circuits and Systems Magazine (2004)
6. Kelly, F.P., Maulloo, A., Tan, D.: Rate control for communication networks: Shadow prices, proportional fairness, and stability. J. Oper. Res. Soc. 49(3), 237–252 (1998)
7. Yang, C., et al.: A taxonomy for congestion control algorithms in packet switching networks. IEEE Network 9 (1995)
8. Low, S.H., Lapsley, D.E.: Optimization Flow Control, I: Basic Algorithm and Convergence. IEEE/ACM Transactions on Networking 7(6), 861–874 (1999)
9. Ogras, U., et al.: Prediction-based flow control for network-onchip traffic. In: Proc. DAC (2006)
10. Hu, J., et al.: DyAD - smart routing for networks-on-chip. In: Proc. DAC (2004)
11. van den Brand, J.W., Ciordas, C., Goossens, K., Basten, T.: Congestion- Controlled Best-Effort Communication for Networks-on-Chip. In: Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE) (April 2007)

12. Talebi, M.S., Jafari, F., Khonsari, A.: Utility-Based Congestion Control for Best Effort Traffic in Network-on-Chip Architecture, (Submitted to MASCOTS 2007) (2007)
13. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge Univ. Press, Cambridge, U.K (2004)
14. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific (1999)
15. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and distributed computation. Prentice-Hall, Englewood Cliffs (1989)
16. Boyd, S.: Convex Optimization II Lecture Notes. Stanford University (2006)
17. Poljak, B.T.: A General Method of Solving Extremum Problems. Soviet Math Doklady 8(3), 593–597 (1967)
18. Grant, M., Boyd, S., Ye, Y.: CVX (Ver. 1.0RC3): Matlab Software for Disciplined Convex Programming, Download available at: http://www.stanford.edu/ boyd/cvx

3D Bathymetry Reconstruction from Airborne Topsar Polarized Data

Maged Marghany, Mazlan Hashim, and Arthur P. Cracknell

Department of Remote Sensing
Faculty of Geoinformation Science and Engineering
Universiti Teknologi Malaysia
81310 UTM, Skudai, Johore Bahru, Malaysia
maged@fksg.utm.my,
magedupm@hotmail.com,
mazlan@fksg.utm.my,
cracknellarthur@hotmail.com

Abstract. This paper introduces a new methods for three-dimensional(3D) ocean bathymetry reconstruction using Airborne TOPSAR Synthetic Aperture data. The new method is based on integration between Fuzzy B-spline and Volterra algorithm. Volterra algorithm is used to simulate the ocean surface current from TOPSAR data. Then, ocean surface current information used as input for continuity equation to estimate the water depths at different locations in TOPSAR data. This study shows that 3D ocean bathymetry can be reconstructed from TOPSAR data. The maximum water depth of 20 m can be captured from TOPSAR data.

Keywords: TOPSAR polarised data, Volterra model, Fuzzy B-spline algorithm, Bathymetry.

1 Introduction

Synthetic aperture radar (SAR) images are the kernel of tremendous number of applications in spite of the noise present on the SAR images. However, scientists and researchers have made excellent efforts to extract and visualize 3D objects in SAR data. In this context, the extraction of 3D data from SAR imaging has been studied by using several techniques such as radargrammetry and interfermetry SAR (InSAR) [24]. However, 3D SAR applications to underwater bathymetry are required certain techniques which are involved integration of the nonlinear model and in situ measurements of sea surface parameters [10]. According to this prospective, several theories concerning the radar imaging mechanism of underwater bathymetry have been established, such as by Alpers and Hennings [1] Shuchman et al[17]; and Vogelzang[20]. The physical theories describing the radar imaging mechanisms for ocean bathymetry are well-understood as three stages: (i) the modulation of the current by the underwater features, (ii) the modulation of the sea surface waves by the variable surface current, and (iii) the interaction of the microwaves with the surface waves[1]. The imaging mechanism which reflects under water topography in a given

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, pp. 410 – 420, 2007.
© Springer-Verlag Berlin Heidelberg 2007

SAR image consists of three models. These models are a flow model, a wave model and the SAR backscatter model. These theories are the basis of commercial services which generate bathymetric charts by inverting SAR images at a significantly lower cost than conventional survey techniques [21]. In this context, Hesselmans et al [8] developed the Bathymetry Assessment System, a computer program which can be used to calculate the depth from any SAR image and a limited number of sounding data points. They found that the imaging model was suitable for simulating a SAR image from the depth map. It showed good agreement between the backscatter in both the simulated and airborne-acquired images, when compared, with accuracy (root mean square) error of $\pm$ 0.23 m within a coastal bathymetry range of 25-30m.

However, the high speckle noise in SAR images has posed great difficulties in inverting SAR images for determining coastal bathymetry. Speckle is a result of coherent interference effects among scatterers which are randomly distributed within each resolution cell. The speckle size, which is a function of the spatial resolution, induces errors in bathymetry signature detections. In order to reduce these speckle effects, appropriate filters, i.e Lee, Gaussian, etc. [12], could be used in the pre-processing stage. The effectiveness of these speckle-reducing filters is however much influenced by local factors and application. Since the SAR images the sea surface, all speckles in SAR images are related to local changes in the surface roughness. This can be due to direct reduction of the wave height (due to slicks), wind induced roughness changes (atmospheric effects) or wave-current interactions (fronts and bathymetry). By contrast, Yu and Scott[19] stated several limitations of the speckle filtering approach. They reported that the size and shape of the filter window can affect the accuracy level of despeckle filters. For instance, too large window size will form a blurred output image, while a small window will decrease the smoothing capability of the filter and will leave speckle. They also found that window shape can lead to changes in the physical characteristics of features in a SAR image. For instance a square window (as is typically applied) will lead to corner rounding of rectangular features. In spite of using despeckle filters to perform edge enhancement, speckle in the neighborhood of an edge (or in the neighborhood of a point feature with high contrast) will remain after filtering. Furthermore, the thresholds used in the enhanced filters, although motivated by statistical arguments, are *ad hoc* improvements that only demonstrate the insufficiency of the window-based approaches. The hard thresholds that enact neighborhood averaging and identity filtering in the extreme cases lead to blotching artifacts from averaging filtering and noisy boundaries from leaving the sharp features unfiltered [19].

In the case of bathymetry mapping not all the filters stated in the literature are suitable for bathymetry application. In fact, SAR data have discontinuities and lower grey levels gradient. Furthermore, by applying some kinds of filter such as Lee, most of bathymetry signature information will be lost. In this context, Inglada and Garello, [9] stated that an anisotropic diffusion filter is more appropriate for speckle reduction in the case of bathymetry signature detection in a SAR image. They concluded that the anisotropic diffusion filter produced the highest smoothed image as the anisotropic diffusion filter preserves the mean grey-level and maintains the bathymetry signature compared to Lee filter. Nevertheless, Inglada and Garello,

[9],[10] were not able to state the accuracy rate of utilizing the Volterra model and anisotropic diffusion filter for SAR bathymetry reconstruction.

In this paper we address the question of reducing the effect of speckle on the accuracy of depth determination in coastal waters using SAR data without needing to include any sounding data values. This is demonstrated with airborne SAR data (namely the TOPSAR) using integration of the Volterra kernel [9] and Fuzzy B-spline models [14]. Four hypotheses examined are: (i) the Volterra model can be used to detect ocean surface current from TOPSAR polarised data, (ii) there are significant differences between the different bands in detecting ocean currents, (iii) the continuity equation can be used to obtain the water depth, and (iv) Fuzzy B-splines can be used to invert the water depth values obtained by the continuity equation into 3-D bathymetry.

2 Methodology

2.1 Data Set

The Jet Propulsion Laboratory (JPL) airborne Topographic Synthetic Aperture Radar (TOPSAR) data were acquired on 6 December 1996 over the coastline of Kuala Terengganu, Malaysia, from 103° 5'E to 103° 9'E and 5° 20'N to 5° 27'N. TOPSAR is a NASA/JPL multi-frequency radar imaging system aboard a DC-8 aircraft and operated by NASA's Ames Research Center at Moffett Field, USA. TOPSAR data are fully polarimetric SAR data acquired with HH-, VV-, HV- and VH-polarized signals from 5m x 5m pixels, recorded for three wavelengths: C band (5 cm), L band (24 cm) and P band (68 cm). A further explanation of TOPSAR data acquisition is given by Melba et al. [15].

2.2 3-D Coastal Water Bathymetry Model

Two models are involved for bathymetric simulation: the Volterra model and the Fuzzy B-spline model. The Volterra model is used to simulate the current velocity from TOPSAR data. The simulation current velocity is used with the continuity equation to derive the water depth variations under different current values. The Fuzzy B-spline is used to reconstruct the two-dimensional water depth to a 3D dimensional.

2.3 Volterra Model

The Volterra model can be used to express the SAR image intensity as a series of nonlinear filters on the ocean surface current. This means that the Volterra model can be used to study the image energy variation as a function of parameters such as the current direction, or the current waveform. A generalized, nonparametric framework to describe the input–output x and y signals relation of a time-invariant nonlinear system is provided by Inglada and Garello[9]. In discrete form, the Volterra series for input, $X(n)$, and output, $Y(n)$ as given by Inglada and Garello [10] can be expressed as:

$$Y(n) = h_0 + \sum_{i_1=1}^{\infty} h_1(i_1)X(n-i_1) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{\infty} h_2(i_1,i_2)X(n-i_1)X(n-i_2) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{\infty}\sum_{i_3=1}^{\infty} h_3(i_1,i_2,i_3)X(n-i_1)X(n-i_2)X(n-i_3) + \ldots\ldots\ldots + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{\infty}\ldots\ldots\ldots\ldots\sum_{i_k=1}^{\infty} h_k(i_1,i_2,\ldots\ldots\ldots,i_k)X(n-i_1)X(n-i_2)\ldots\ldots\ldots\ldots X(n-i_k) \quad (1)$$

where, n, i_1 , i_2 ,...,i_k , are discrete time lags. The function $h_k(i_1$,i_2 ,...,$i_k)$ is the kth-order Volterra kernel characterizing the system. The h_1 is the kernel of the first order Volterra functional, which performs a linear operation on the input and h_2 , h_3 ,...,h_k capture the nonlinear interactions between input and output TOPSAR signals. The order of the non-linearity is the highest effective order of the multiple summations in the functional series.

Following Inglada and Garello [9], the modification of the Volterra kernels $H_{1y}(v_x, v_y)$ of the frequency domain for the current flow in the range direction can be described as:

$$H_{1y}(v_x, v_y) = k_y \langle \vec{U} \rangle \vec{v} \left[\vec{K}^{-1} \left[\frac{\partial \psi}{\partial \vec{k}} + \frac{\partial \vec{c}_g}{\partial x}\vec{v} + \frac{\partial \vec{u}_y}{\partial y} v_y + 0.5u_x \right] \right] \left[\frac{\partial \psi}{\partial \omega} \right] \frac{\vec{c}_g(\vec{K})\vec{v} + j.0.5\omega_0^{-1}}{[\vec{c}_g(\vec{K})\vec{v}]^2 + [0.5\omega_0^{-1}]^2} + j(6.10^{-3}.\vec{K}^{-4})(\frac{R}{V})v_x \quad (2)$$

where $\langle \vec{U} \rangle$ is the mean current velocity, $\vec{u}_y$ is the current flow along the range direction while $\vec{u}_x$ is current gradient along the azimuth direction. k_y is the wave number along the range direction, $\vec{K}$ is the spectra wave vector, ω_0 is the angular wave frequency, $\vec{c}_g$ is the group velocity, ψ is the wave spectra energy, v stands for the Volterra kernel frequency along the azimuth and range directions and R/V is the range to platform velocity ratio, in the case of TOPSAR 32 s.

According to Vogelzang et al. [21],[22] the current movement along the range direction can be estimated by

$$\vec{U}_y(v_x, v_y) = \frac{FT\left[\prod_{j=1}^{i} x(t)\right]}{H_{1y}(v_x, v_y)} \quad (3)$$

where $FT\left[\prod_{j=1}^{i} x(t)\right]$ is the linearity of the Fourier transform for the input TOPSAR image intensity. The inverse filter $P(v_x, v_y)$ is used since $H_{1y}(v_x, v_y)$ has a zero

for (v_x, v_y) which indicates that the mean current velocity should have a constant offset. The inverse filter $P(v_x, v_y)$ can be given as

$$P(v_x, v_y) = \begin{cases} [H_{1y}(v_x, v_y)]^{-1} & if\,(v_x, v_y) \neq 0, \\ 0 & otherwise. \end{cases} \tag{4}$$

Then, the continuity equation is used to estimate the water depth as given by Vogelzang et al. [22]

$$\frac{\partial \zeta}{\partial t} + \nabla .\{(h + \zeta)\vec{U}_y\} = 0 \tag{5}$$

where ζ is the surface elevation above the mean sea level, which is obtained from the tidal table, *t* is the time and *h* is the local water depth. The real current data was estimated from the Malaysian tidal table of 6 December, 1996.

2.4 The Fuzzy B-splines Method

The fuzzy B-splines (FBS) are introduced allowing fuzzy numbers instead of intervals in the definition of the B-splines. Typically, in computer graphics, two objective quality definitions for fuzzy B-splines are used: triangle-based criteria and edge-based criteria. A fuzzy number is defined using interval analysis. There are two basic notions that we combine together: confidence interval and presumption level. A confidence interval is a real values interval which provides the sharpest enclosing range for current gradient values. An assumption level μ -level is an estimated truth value in the [0,1] interval on our knowledge level of the gradient current [2],[3],[4]. The 0 value corresponds to minimum knowledge of gradient current, and 1 to the maximum gradient current. A fuzzy number is then prearranged in the confidence interval set, each one related to an assumption level $\mu \in [0,1]$. Moreover, the following must hold for each pair of confidence intervals which define a number: $\mu \succ \mu' \Rightarrow h \succ h'$. Let us consider a function $f : h \to h'$, of N fuzzy variables $h_1, h_2, ..., h_n$. Where h_n are the global minimum and maximum values of the water depth of the function on the current gradient along the space. Based on the spatial variation of the gradient current, and water depth, the fuzzy B-spline algorithm is used to compute the function *f*. The construction begins with the same pre-processing aimed at the reduction of measured current values into a uniformly spaced grid of cells. As in the Volterra model, data are derived from the TOPSAR polarised backscatter images by the application of a 2-dimensional fast Fourier transform. First of all, each estimated current data value in a fixed kernel window size of 512x512 pixels and lines is considered as a triangular fuzzy number defined by a minimum, maximum and measured value. Among all the fuzzy numbers falling within a kernel window size, a fuzzy number is defined whose range is given by the minimum and maximum values of gradient current and water depth along each kernel window size. Furthermore, the identification of a fuzzy number is acquired to summarize the

estimated water depth data in a cell and it is characterized by a suitable membership function. The choice of the most appropriate membership is based on a triangular number whose support is the range of water depth data in the cell and whose vertex is the median value of water depth data[4]. Fig. 1 shows the regions of interest that were used to simulate the bathymetric information from L-band with HH polarization. The bathymetry information has been extracted from 4 sub-images, where each sub-image was 512 x 512 pixels.

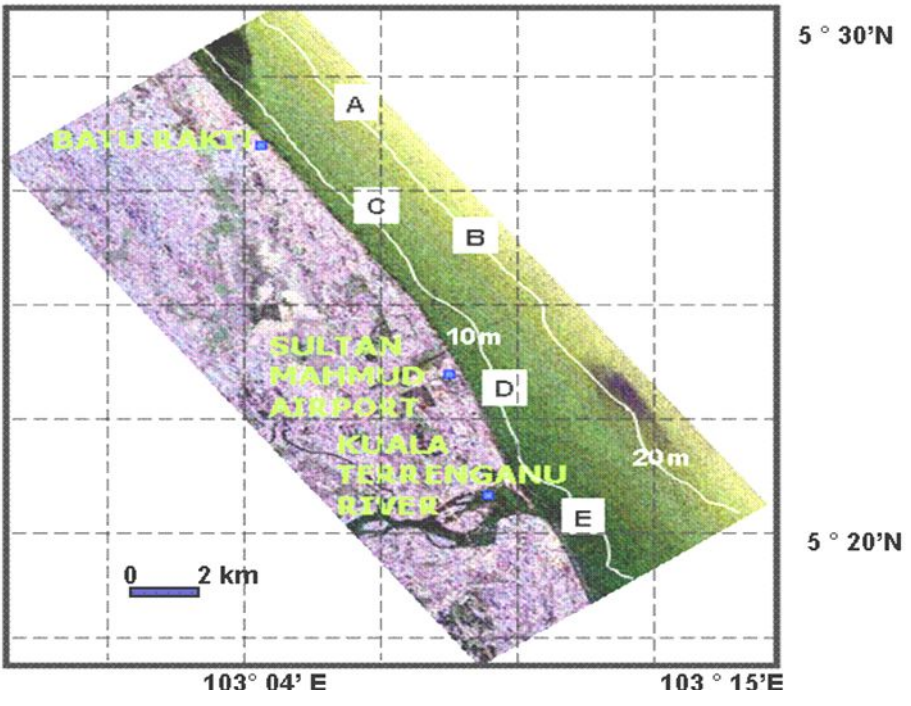


Fig. 1. Selected window sizes of A to D with 512 x 512 pixels

In order to evaluate the simulation method quantitatively, the regression model and root mean square were computed for the simulated bathymetry from TOPSAR data and bathymetry points extracted from a bathymetric chart of 1998, sheet number 4365 of 1:25,000 scale.

3 Results and Discussion

Fig. 2 shows the signature of the underwater topography. Underwater topography is obvious as frontal lines parallel to the shoreline. This is due to the fact that the ocean signature of the boundary is clear in the brightness of a radar return, since the backscatter tends to be proportional to wave height [22]. In C-band with VV polarization, this feature is clearly weaker than at L –band with HH polarization. It is

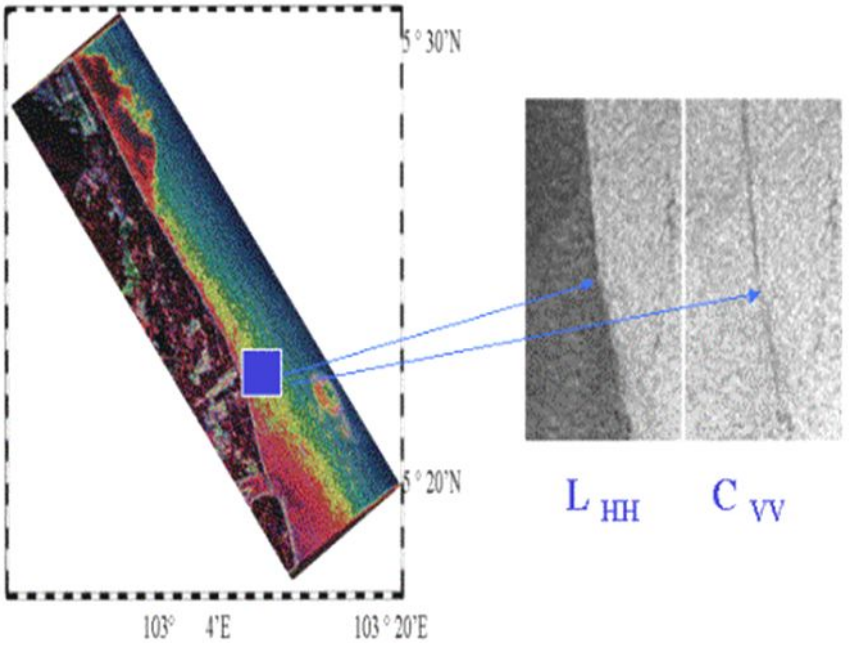


Fig. 2. Bathymetry signature with different bands

possible that the character of the current gradient is such that the L_{HH}-band surface Bragg waves are more strongly modulated than for C_{VV} band. This may provide an explanation for weaker bathymetric signatures at C_{VV} band. The finding is similar to that of Romeiser and Alpers [17].

Fig. 3 shows the comparison between the 3-D bathymetry reconstruction from the topographic map, the L_{HH} band data, and the C_{VV} band data. It is obvious that the coastal water bathymetry along the Sultan Mahmud Airport has a gentle slope and the bathymetric contours are parallel to the shoreline. Close to the river mouth, the bathymetry at this location shows a sharp slope [13]. The L_{HH} band captured a more real bathymetry pattern than the C_{VV} band. This result could be confirmed with a regression model in Fig.4. Fig. 4a shows the regression relation between the observed bathymetry and the results obtained using the C_{VV} band TOPSAR data. Fig.4b shows a similar regression relation for L_{HH} TOPSAR data. The scatter points in Figure 4b are more close to the regression line than those in Fig. 4a. The bathymetry simulation from L_{HH} band with r^2 value of 0.677. It might be that the HH polarization has a larger tilt modulation compared to the VV polarization. Tilt modulation explains that the Bragg scattering is dependent on the local incident angle. The long wavelength of L-band HH polarization modulate this angle, hence modifying the Bragg resonance wave length. It might be due to the fact that the first – order Bragg Scattering gives good results for long radar wavelengths (L-band), but for shorter radar wavelength (C-band) the effects of waves longer than the Bragg waves must be taken into account [17],[18]. This could be due to strong current flow from the mouth river of the Kuala Terengganu. This study confirms the study of Maged [13],[14].

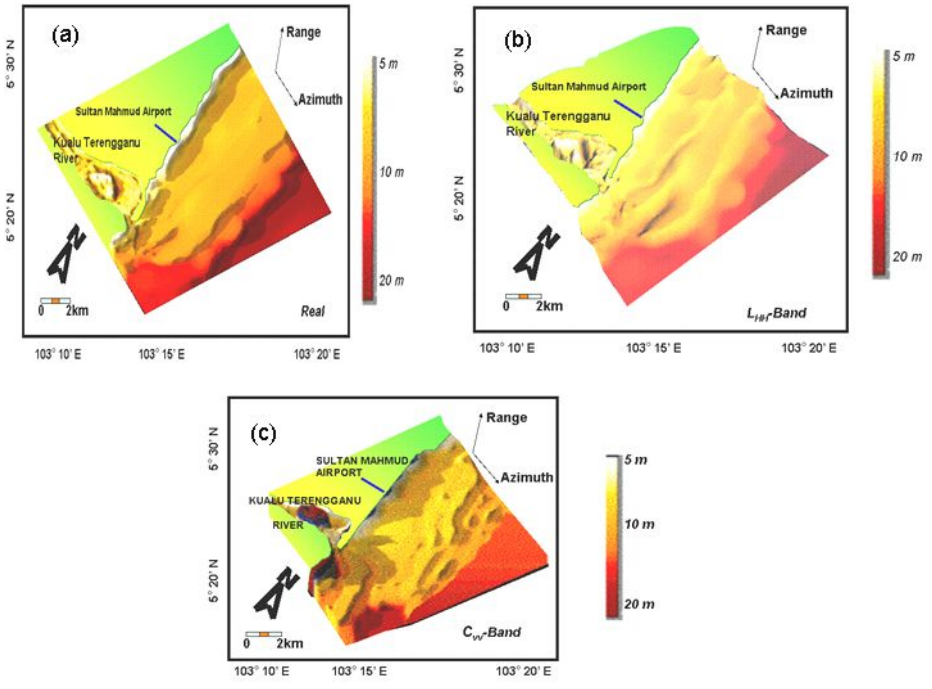


Fig. 3. Three-dimensional bathymetry reconstructions from (a) real topography map (b) L_{HH} Band and (c) C_{VV} Band

The visualization of 3-D bathymetry is sharp with the different TOPSAR polarised bands and real data due to the fact that each operation on a fuzzy number becomes a sequence of corresponding operations on the respective μ-levels , and the multiple occurrences of the same fuzzy parameters evaluated as a result of the function on fuzzy variables [6],[7],[8]. It is very easy to distinguish between smooth and jagged bathymetry. Typically, in computer graphics, two objective quality definitions for fuzzy B-splines were used: triangle-based criteria and edge-based criteria. Triangle-based criteria follow the rule of maximization or minimization, respectively, of the angles of each triangle [6],[8],[9]. The so-called max-min angle criterion prefers short triangles with obtuse angles. This finding confirms those of Keppel [11],[5]. The distinction between smooth and rough bathymetry can be seen in Figure 6 where the symmetric three-dimensional structure of the bathymetry of a segment of a connecting depth. Smooth sub-surfaces appear in Fig.3 where the near-shore bathymetric contour of 5 m water depth runs nearly parallel in 3D-space to the coastline. A rough sub-surface structure appears in steep regions of 20 m water depth. This is due to the fact that the fuzzy B-splines considered as deterministic algorithms which are described here optimize a triangulation only locally between two different points [2]. This corresponds to the feature of deterministic strategies of finding only sub-optimal solutions usually. The three-dimensional bathymetry construction is not similar to the study of Inglada and Garello [9], such that in the latter the bathymetry was

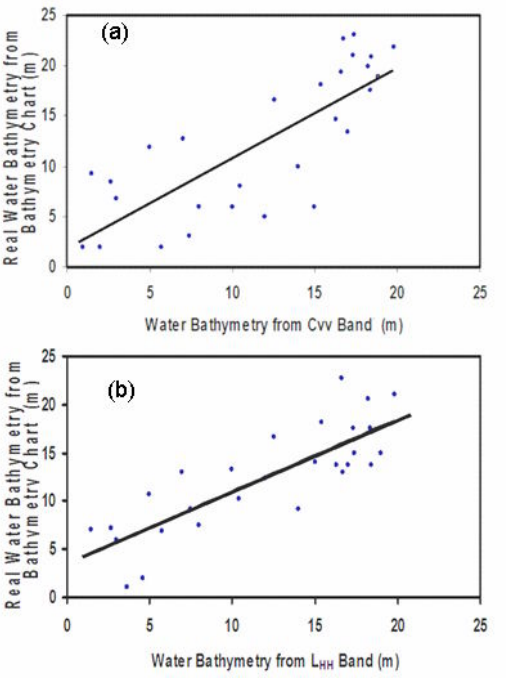


Fig. 4. Regression model between real water bathymetry from bathymetry chart and (a) water bathymetry from C_{VV} Band ($r^2=0.62$; $y=0.89x+1.89$) and (b) L_{HH} Band ($r^2=0.68$;$y=0.75+3.46$)

constructed in the shallow sand waves due to the limitation of the inversion of the linear kernel of the Volterra model. The integration of the inversion of the Volterra model with fuzzy B-splines improved the three-dimensional bathymetry reconstruction pattern.

The accuracy standards for hydrographical charting with acoustic or mechanical based sounding defined as the total error of derived depths, should not exceed, with probability of 90%, $\pm$0.3m for depths less than 30m or 1% of the depths greater than 30m[16],[23]. This is in fact a very stringent requirement, and the derived bathymetric information from non-contact techniques including laser bathymetric sensors [7] which at present are very much dependent on the clarity of water for achieving high accuracy as stipulated in International Hydraulic Organization (IHO). However for a general survey to identify moving shoals within navigational channel or coasts, for large aerial reconnaissance, this technique demonstrates its potential at relatively low cost before detailed surveys can be carried out. In the case of this study, the TOPSAR data was acquired at 10m resolution, adequate only for extracting bathymetric information with 7.23m accuracy. Based on Forster [5],[6], the relationship of spatial resolution of data and the level of three-dimensional information that can be determined it is $P_m \leq 3x10^{-4} S_m$; where P_m is pixel size of the satellite data, and S_m is the map scale. For 10m pixel size this implies a scale of around 1:30,000, for which the smallest mmu is around $\pm$15m based on the pointing accuracy of 0.5mm on the map scale. In this context, an error of $\pm$ 7.2 m in depth was acquired with L_{HH} band

with 10 m spatial resolution of TOPSAR data. Furthermore, high spatial resolution data with 1 m could potentially improve the bathymetric map with an error of $\pm$0.7m.

4 Conclusions

This work has demonstrated the three-dimensional bathymetry reconstruction from TOPSAR polarized data. The Volterra and fuzzy-B splines algorithms are used to reconstruct the bathymetry pattern The L_{HH} band provides an better approximation to the real bathymetry than does the C_{VV} band. It can be said that the integration between the Volterra model and the fuzzy B-splines could be an excellent tool for 3-D bathymetry determination from SAR data.

References

1. Alpers, W., Hennings, I.: A theory of the imaging mechanism of underwater bottom topography by real and synthetic aperture radar. Journal of Geophysical Research 89, 10529–10546 (1984)
2. Anile, A.M.: Report on the activity of the fuzzy soft computing group, Technical Report of the Dept. of Mathematics, University of Catania, p. 10 (March 1997)
3. Anile, A.M., Deodato, S., Privitera, G.: Implementing fuzzy arithmetic. Fuzzy Sets and Systems 72, 123–156 (1997)
4. Anile, A.M., Gallo, G., Perfilieva, I.: Determination of Membership Function for Cluster of Geographical data. Genova, Italy: Institute for Applied Mathematics, National Research Council, University of Catania, Italy, October 1997. vol. 26/97 p. 25 Technical Report (1997)
5. Forster, B.C.: Mapping Potential of Future Spaceborne Remote Sensing System. In: Procs of 27th Australia Survey Congress, Alice Springs, Institution of Surveyors, Australia, Australia, pp. 109–117 (1985)
6. Fuchs, H., Kedem, Z.M., Uselton, S.P.: Optimal Surface Reconstruction from Planar Contours. Communications of the ACM 20, 693–702 (1997)
7. Guenther, G.C., Cunningham, A.G., LaRocque, P.E., Reid, D.J.: In: Proceedings of EARSeL-SIG-Workshop LIDAR,Dresden/FRG,EARSeL, Strasbourg, France (June 16-17, 2000)
8. Hesselmans, G.H, Wensink, G.J., Koppen, C.G.V., Vernemmen, C.: Bathymetry assessment Demonstration off the Belgian Coast-Babel. The Hydrographica Journal 96, 3–8 (2000)
9. Inglada, J., Garello, R.: Depth estimation and 3D topography reconstruction from SAR images showing underwater bottom topography signatures. In: Proceedings of Geoscience and Remote Sensing Symposium, 1999, IGARSS'99, Hamburg, Germany, 28 June-2 July 1999, IEEE Geoscience and Remote Sensing Society, USA. vol. 2, pp. 2956–2958 (1999)
10. Inglada, J., Garello, R.: On rewriting the imaging mechanism of underwater bottom topography by synthetic aperture radar as a Volterra series expansion. IEEE Journal of Oceanic Engineering 27, 665–674 (2002)
11. Keppel, E.: Approximation Complex Surfaces by Triangulations of Contour Lines. IBM Journal of Research Development 19, 2–11 (1975)

12. Lee, J.S., Schuler, D., Ainsworth, T.L., Krogager, E., Kasilingam, D., Boerner, W.M.: On the estimation of radar polarization orientation shifts induced by terrain slopes. IEEE Transactions on Geosciences and Remote Sensing 40, 30–41 (2002)
13. Maged, M.: Coastal Water Circulation off Kuala Terengganu, Malaysia. MSc. Thesis Universiti Pertanian Malaysia (now Universiti Putra Malaysia) (1994)
14. Maged, M.: Fuzzy B-spline and Volterra algorithms for modelling surface current and ocean bathymetry from polarised TOPSAR data. Asian Journal of Information Technology 4, 1–6 (2005)
15. Melba, M., Kumar, S., Richard, M.R., Gibeaut, J.C., Amy, N.: Fusion of Airborne polarmetric and interferometric SAR for classification of coastal environments. IEEE Transactions on Geoscience and Remote Sensing 37, 1306–1315 (1999)
16. Mills, G.B.: NOAA, Office of Coast Survey, Hydrographic Surveys Division, 1315 East-West Highway, Station 6859, Silver Spring, Maryland, USA 20910-3282 (accessed December 2006) (Url: http://chartmaker.ncd.noaa.gov/hsd/ihr-s44.pdf
17. Romeiser, R., Alpers, W.: An improved composite surface model for the radar backscattering cross section of the ocean surface, 2, Model response to surface roughness variations and the radar imaging of underwater bottom topography. Journal of Geophysical Research 102, 25,251–25,267 (1997)
18. Shuchman, R.A., Lyzenga, D.R., Meadows, G.A.: Synthetic aperture radar imaging of ocean-bottom topography via tidal-current interactions: theory and observations. International Journal of Remote Sensing 6, 1179–1200 (1985)
19. Yu, Y., Scott, T.A.: Speckle reducing anisotropic diffusion. IEEE Transactions on Geoscience and Remote Sensing 11, 1260–1270 (2002)
20. Vogelzang, J.: Mapping submarine sand waves with multiband imaging radar, 1, Model development and sensitivity analysis. Jornal of Geophysical Research 102, 1163–1181 (1997)
21. Vogelzang, J., Wensink, G.J., Calkoen, C.J., van der Kooij, M.W.A.: Mapping submarine sand waves with multiband imaging radar, 2, Experimental results and model comparison. Journal of Geophysical Research 102, 1183–1192 (1997)
22. Vogelzang, J., Wensink, G.J., de Loor, G.P., Peters, H.C., Pouwels, H.: Sea bottom topography with X band SLAR: the relation between radar imagery and bathymetry. International Journal of Remote Sensing 13, 1943–1958 (1992)
23. Wensink, H., Campbell, G.: Bathymetric map production using the ERS SAR. Backscatter 8(1), 17–22 (1997)
24. Homer, J., Longstaff, D., She, Z.: Improved digital elevation models via multi-baseline interferometric SAR. In: Proceedings of IGARSS, pp.1579–1581 (1997)

A Parallel FDTD Algorithm for the Solution of Maxwell's Equations with Nearly PML Absorbing Boundary Conditions

Omar Ramadan

Department of Computer Engineering,
Eastern Mediterranean University,
Gazimagusa, Mersin 10,
Turkey
omar.ramadan@emu.edu.tr

Abstract. A parallel three dimensional (3-D) finite difference time domain (FDTD) algorithm for the solution of Maxwell's equations with nearly perfectly matched layer (NPML) absorbing boundary conditions is presented. In the proposed algorithm, the FDTD computational domain is divided into subdomains using two-dimensional topology and the interprocessor communication operations between the neighboring subdomains are carried out by using the message passing interface (MPI) library. The performance of the proposed parallel algorithm has been studied by using a point source radiating in 3-D domains. It has been observed that the proposed parallel system provides a speed up factor of 13.39 with 16 PCs for $240 \times 240 \times 40$ domain size.

1 Introduction

The finite difference time domain (FDTD) [1] method has been widely used for solving electromagnetic problems [2]. The main advantage of the FDTD method is that it is a straightforward solution of the six-coupled field components of the Maxwell's curl equations. This method is based on Yee's algorithm [1] and computes the electric and the magnetic field components by discretizing the Maxwell's curl equations both in time and space, and then solving the discretized equations in a time marching sequence by alternatively calculating the electric and magnetic fields in the computational domain.

An important issue in the FDTD research is the requirements of efficient absorbing boundary conditions (ABCs) to truncate open region problems. Berenger's perfectly matched layer (PML) has been shown to be one of the most effective FDTD ABCs [3]. Berenger's PML consists of a lossy and artificial layer with a thickness varying from 4 to 12 cells and can be placed generally very close to the radiating structures located inside the inner FDTD computational domain. This type of PML is based on splitting each field component into two subcomponents.

To analyze large problems using the FDTD method, intensive computing time and memory storage are needed. Hence, parallelizing the FDTD algorithm has

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 421–430, 2007.
© Springer-Verlag Berlin Heidelberg 2007

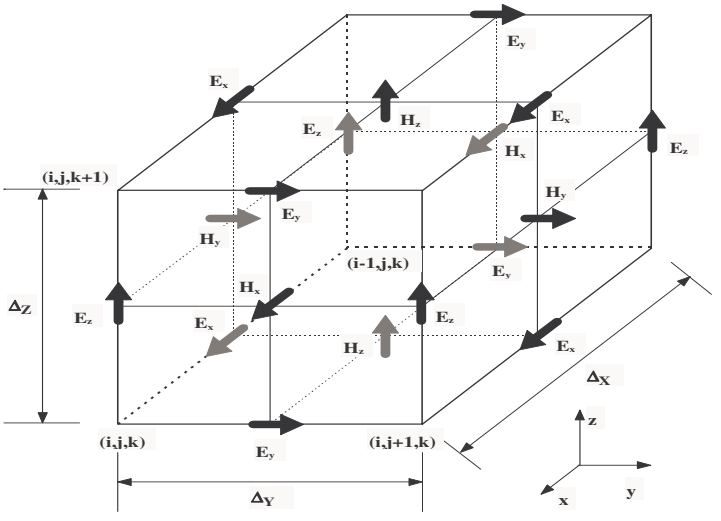


Fig. 1. Yee's unit cell

been shown to be one of the latest challenges in the FDTD research. Recently, new parallel PML algorithms, based on the message passing interface (MPI) system [4], [5], has been introduced for truncating open region domains [6], [7].

In this paper, an alternative parallel PML algorithm, based on the nearly PML (NPML) [8], is introduced. The proposed parallel algorithm has the same number of interprocessor communication operations as the parallel implementations of the conventional FDTD algorithm. Therefore, the proposed algorithm can be applied in the NPML region at the domain boundaries as well as the inner FDTD computational domain. In the proposed parallel algorithm, the computational domain is divided into subdomains by using two-dimensional topology. The validity of the proposed parallel algorithm is shown through numerical simulations carried out for a point source radiating in 3-D domains and performed on a network of PCs interconnected with ethernet. It has been observed that the proposed parallel algorithm provides a speed up factor of 13.39 with 16 PCs for $240 \times 240 \times 40$ domain size.

The paper is organized as follows. In sections 2 and 3, the formulations of the FDTD and the NPML algorithms are presented, respectively. In section 4, the proposed parallelization technique is described. Section 5 includes the results to validate the proposed parallel algorithm. Finally, a summary and conclusions are included in section 6.

2 FDTD Formulations

In a linear, homogeneous, isotropic and lossless medium, the Maxwell's equations can be written in the frequency domain as

$$\nabla \times \mathbf{H} = j\omega\varepsilon_0 \mathbf{E} \tag{1}$$

$$\nabla \times \mathbf{E} = -j\omega\mu_0 \mathbf{H} \tag{2}$$

where c is the speed of light. Equations (1) and (2) can be decomposed into a system of six first order scalar differential equations in terms of the fields E_x, E_y, E_z, H_x, H_y, and H_z. As an example, E_z field component can be written as

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon_0}\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y}\right) \tag{3}$$

Using the Yee's FDTD algorithm [1], the six field components are organized in the Yee's unit cell as shown in Fig. 1: the electric field components lies on the center of the edges of the Yee's cell and computed at the time $t = (n+1)\Delta t$, where Δt is the time step. On the other hand, the magnetic fields are located at the center of each face of the Yee's unit cell and computed at the time $t = (n+1/2)\Delta t$. Applying central-differencing discretization in both space and time to the continuous differential operators of (3), E_z can be computed as [1]

$$E^{n+1}_{z_{i,j,k+1/2}} = E^{n}_{z_{i,j,k+1/2}} + \chi\left[H^{n+1/2}_{y_{i+1/2,j,k+1/2}} - H^{n+1/2}_{y_{i-1/2,j,k+1/2}} - H^{n+1/2}_{x_{i,j+1/2,k+1/2}} + H^{n+1/2}_{x_{i,j-1/2,k+1/2}}\right] \tag{4}$$

where $\Delta = \Delta x = \Delta y = \Delta z$ is the space cell size in the x, y, and z directions, and $\chi = \Delta t/\varepsilon_0 \Delta$. Similar formulations can be obtained for the other field components.

3 NPML Formulation

Using the NPML formulations [8], the normalized frequency domain modified Maxwell's equations in the PML region at the domain boundaries can be written as

$$\widetilde{\nabla} \times \mathbf{H} = j\omega\varepsilon_0 \mathbf{E} \tag{5}$$

$$\widetilde{\nabla} \times \mathbf{E} = -j\omega\mu_0 \mathbf{H} \tag{6}$$

where

$$\widetilde{\nabla} = \sum_{\eta=x,y,z} \widehat{a}_\eta \frac{\partial}{\partial \eta}\frac{1}{S_\eta} \tag{7}$$

where $S_\eta(\eta = x, y, z)$ is the NPML stretched coordinate variables in the $\eta-$direction and chosen as

$$S_\eta = 1 + \sigma_\eta / j\omega\varepsilon_o \tag{8}$$

where σ_η is the conductivity profile in the NPML region along the $\eta-$direction. It is important to not that when the NPML conductivity profiles are zero, (5)

and (6) will be reduced to (1) and (2). To discretize (5) and (6), consider as an example, the E_z-field component of (5):

$$j\omega\widetilde{E}_z=\frac{1}{\varepsilon_0}\left(\frac{\partial}{\partial x}\frac{\widetilde{H}_y}{S_x}-\frac{\partial}{\partial y}\frac{\widetilde{H}_x}{S_y}\right) \tag{9}$$

where $\widetilde{E}_z$, $\widetilde{H}_x$, and $\widetilde{H}_y$ are the Fourier transform of the corresponding fields. Equations (9) can be written as [8]

$$j\omega\widetilde{E}_z=\frac{1}{\varepsilon_0}\left(\frac{\partial}{\partial x}\widetilde{H}_{yx}-\frac{\partial}{\partial y}\widetilde{H}_{xy}\right) \tag{10}$$

where the auxiliary variables $\widetilde{H}_{yx}$ and $\widetilde{H}_{xy}$ are given by

$$\widetilde{H}_{yx}=\frac{j\omega}{j\omega+\sigma_x/\varepsilon_o}\widetilde{H}_y \tag{11}$$

$$\widetilde{H}_{xy}=\frac{j\omega}{j\omega+\sigma_y/\varepsilon_o}\widetilde{H}_x \tag{12}$$

Transforming (10) into the time domain using the inverse Fourier transform relation $j\omega\longrightarrow\partial/\partial t$, we obtain

$$\frac{\partial}{\partial t}E_z=\frac{1}{\varepsilon_0}\left(\frac{\partial}{\partial x}H_{yx}-\frac{\partial}{\partial y}H_{xy}\right) \tag{13}$$

where $\widetilde{H}_{yx}$ and $\widetilde{H}_{xy}$ are given by the following first order differential equations obtained by transforming (11) and (12) into the time domain:

$$\frac{\partial}{\partial t}H_{yx}+\frac{\sigma_x}{\varepsilon_o}H_{yx}=\frac{\partial}{\partial t}H_y \tag{14}$$

$$\frac{\partial}{\partial t}H_{xy}+\frac{\sigma_y}{\varepsilon_o}H_{xy}=\frac{\partial}{\partial t}H_x \tag{15}$$

By following the conventional Yee's FDTD algorithm for discretizing the space and the time derivatives in (13), the following FDTD expression to calculate the E_z field can be obtained:

$$\begin{aligned}E^{n+1}_{z_{i,j,k+1/2}}=E^{n}_{z_{i,j,k+1/2}}&+\chi(H^{n+1/2}_{yx_{i+1/2,j,k+1/2}}-H^{n+1/2}_{yx_{i-1/2,j,k+1/2}})\\&-\chi(H^{n+1/2}_{xy_{i,j+1/2,k+1/2}}-H^{n+1/2}_{xy_{i,j-1/2,k+1/2}})\end{aligned} \tag{16}$$

where the additional auxiliary variables are obtained by using the standard Yee's FDTD algorithm for discretizing (14) and (15) as,

$$H^{n+1}_{yx_{i,j,k+1/2}}=r_{0_{i+1/2}}H^{n}_{yx_{i,j,k+1/2}}+r_{1_{i+1/2}}(H^{n+1/2}_{y_{i+1/2,j,k+1/2}}-H^{n-1/2}_{y_{i+1/2,j,k+1/2}}) \tag{17}$$

$$H^{n+1}_{xy_{i,j,k+1/2}}=r_{0_{j+1/2}}H^{n}_{xy_{i,j,k+1/2}}+r_{1_{j+1/2}}(H^{n+1/2}_{x_{i,j+1/2,k+1/2}}-H^{n-1/2}_{x_{i,+-1/2,k+1/2}}) \tag{18}$$

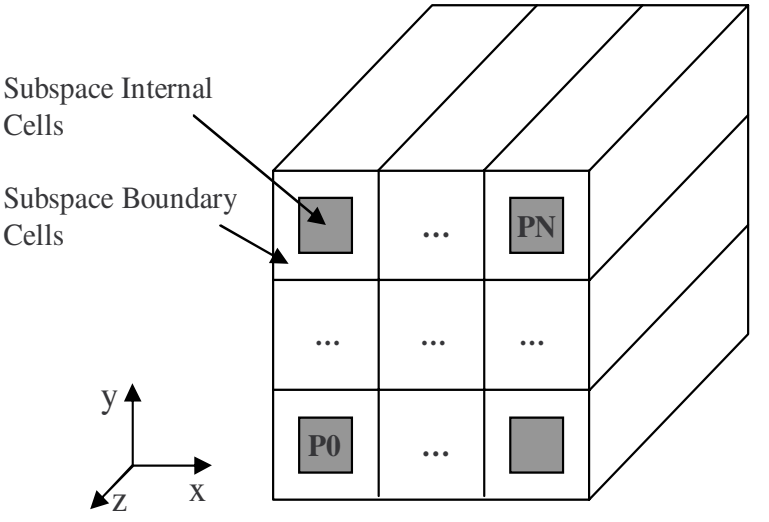


Fig. 2. Computational domain partitioning

where

$$r_{0_l} = \frac{1-\alpha_l}{1+\alpha_l}, \text{ and } r_{1_l} = \frac{1}{1+\alpha_l}, \text{ for } l = i, j \tag{19}$$

with

$$\alpha_l = \frac{\Delta t \sigma_l}{2\varepsilon_o} \tag{20}$$

Similar expressions can be obtained for the other field components.

4 Parallelizing Strategy

Using the two-dimensional topology, the computational domain is divided into subdomains along the xy directions, where each subdomain is assigned to one processor, as shown in Fig. 2. To calculate the field components at the subdomain boundaries, data from the neighboring subdomains are needed. In this paper, the MPI system is used to exchange data between processors. A complete detail of the MPI is provided in [4], [5]. Fig. 3 shows the data to be exchanged between neighboring subdomains in order to parallelize the proposed NPML formulations. It is interesting to note that the proposed NPML parallel algorithm involves the same number of interprocessor communication operations as the conventional FDTD algorithm. Hence, the proposed algorithm can be applied in the NPML region as well as in the inner FDTD domain. Based on Fig. 3, the parallel implementation of the NPML algorithm can be summarized as:

1. MPI initialization.
2. Reading the simulation parameters.

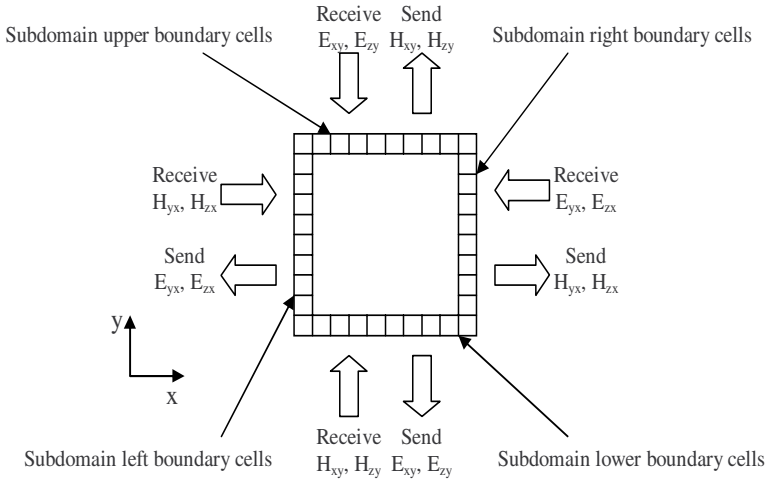


Fig. 3. Communications at the boundaries of a subdomain for the proposed parallel PML algorithm

3. Divide the computational domain into subdomains.
4. At each time step:
  4.1 Calculate E_x, E_y and E_z field components and the additional auxiliary variables.
  4.2 Communicate E_{yx}, E_{zx}, E_{xy}, E_{zy}at the subdomain boundaries.
  4.3 Calculate H_x, H_y and H_z field components and the additional auxiliary variables.
  4.4 Communicate H_{yx}, H_{zx}, H_{xy}, H_{zy} at the subdomain boundaries.
5. MPI finalization.

5 Simulation Study

The validity of the proposed formulations is shown through numerical examples carried out for a radiation problem. In this test, a point source excites a 3-D domain at its center. The cell size was chosen as $\Delta = \Delta x = \Delta y = \Delta z = 15mm$ and the time step was taken as $\Delta t = 25ps$. The excitation pulse used in this test was chosen to be similar to the pulse used in [9] and it is defined as

$$E_z^n = \begin{cases} \alpha(10 - 15\cos\omega_1\xi + 6\cos\omega_2\xi - \cos\omega_3\xi) & \xi \leq \tau \\ 0 & \xi > \tau \end{cases} \tag{21}$$

where $\alpha = \frac{1}{32}$, $\tau = 10^{-9}s$, $\xi = n\Delta t$, and $\omega_m = \frac{2\pi m}{\tau}$, $m = 1, 2, 3$. The computational domain is truncated by 8 additional PML layers with a quadratic conductivity profile and with a theoretical reflection coefficient 10^{-5}, i.e., PML[8, 2, 0.001%] as defined in [3]. The proposed parallel system consists of 4, 8 or 16

Table 1. Parallel FDTD System

CPU	Pentium IV 2.2 GHz
Memory	512 Mbyte
Processor number	4, 8, or 16
Communication software	Message passing interface
Network interface	100Mbps
Operating system	Windows XP
Compiler	C++

Table 2. Simulation time (in seconds) for the proposed parallel system

Grid Size (cells)	Number of PCs			
	1 PCs	4 PCs	8 PCs	16 PCs
$240 \times 240 \times 40$	714.31	185.27	100.43	53.31
$360 \times 360 \times 40$	N/A	412.82	229.54	120.3
$480 \times 480 \times 40$	N/A	739.82	379.77	200.33

PCs interconnected with 100 Mbps network interface. Table 1 shows the characteristics of the proposed parallel system.

Table 2 shows the simulation time as obtained by using one, four, eight and sixteen PCs for three different domain sizes. Due to the memory limitations of the PCs used in this study, the results of the serial code for large domain sizes are not available and represented by N/A. Figure 4 shows the total simulation time of the proposed parallel algorithms as a function of the number of cells in the domains, i.e., $N_x \times N_y \times N_z$. As can be seen from Table 2 and Fig. 4, the proposed parallel algorithm provides a significant reduction in the simulation time compared with the serial NPML.

To measure the performance of the parallel algorithm, the speedup was calculated as

$$S(N) = T(1)/T(N) \tag{22}$$

where $T(1)$ is the time needed to solve the problem using one PC and $T(N)$ is the time needed to solve the same problem using N PCs. Figure 5 shows the speedup obtained with four, eight and sixteen PCs for $240 \times 240 \times 40$ domain size. For other domain sizes, the speedup is not calculated as the serial results are not available due to memory limitations of the PCs. For the purpose of comparison, the ideal speedup is also shown in Fig. 5. As can be seen from Fig. 5, with 16 PCs parallel system the proposed parallel algorithm provides a speedup factor of 13.39 for the $240 \times 240 \times 40$ domain size. Figure 6 shows the relative speed up obtained with 8 and 16 PCs parallel systems as compared with the 4 PCs parallel system as a function of the number of cells in the domain. As can be seen from Fig. 6, with 8 PCs parallel system, a relative speedup factor of 1.95 was achieved for the $240 \times 240 \times 40$ domain size while a relative speedup factor of 3.69 was achieved with the 16 PCS parallel system. Therefore,

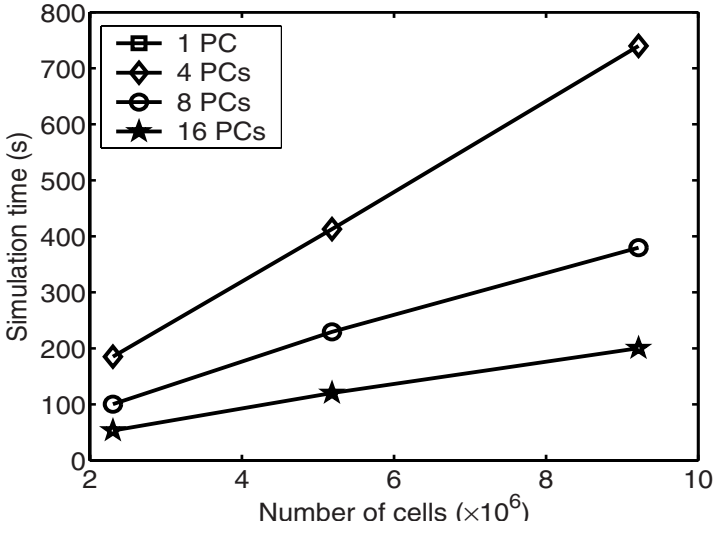


Fig. 4. Simulations time for the proposed parallel NPML algorithm

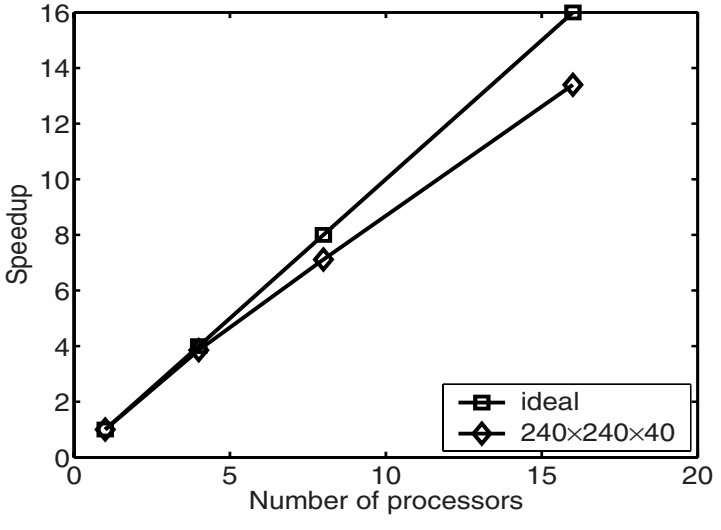


Fig. 5. Speed-up for the proposed parallel NPML algorithm

it must be mentioned that when the computational domains are partitioned over many processors, the efficiency of the parallelization will reach a limitation. This is because the communication time needed to perform the data exchange between the processors becomes comparable to the computational time needed to compute the field components.

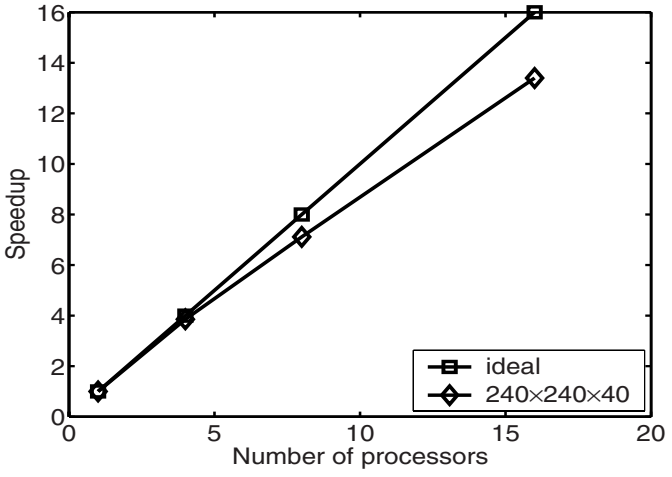


Fig. 6. Relative speed of 8 and 16 PCs parallel system as compared with 4 PCs parallel system

6 Conclusion

In this paper, a parallel NPML algorithm, based on the MPI library, has been implemented for truncating 3-D FDTD problems. The performance of the proposed parallel algorithm has been studied by using a point source radiating in 3-D domains. It has been observed that a speedup factor of 13.39 has been obtained with 16 PCs parallel system for the $240 \times 240 \times 40$ domain size.

References

1. Yee, K.S.: Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. IEEE Transaction on Antennas and Propagation 14, 302–307 (1966)
2. Taflove, A.: Computational electrodynamics: The Finite-Difference Time-Domain Method. Artech House, Boston, London (1995)
3. Berenger, J.P.: A perfectly matched layer for the absorption of electromagnetic waves. Journal of Computational Physics 114, 185–200 (1994)
4. Gropp, W., Lusk, E., Skjellum, A.: Using MPI: Potable parallel Programming with the Message-Passing Interface. MIT Press, Cambridge (1994)
5. Pacheco, P.S.: Parallel Programming with MPI. Morgan Kaufmann Publishers, San Francisco (1997)
6. Hoteit, H., Sauleau, R., Philippe, B., Coquet, P., Daniel, J.P.: Vector and parallel implementations for the FDTD analysis of millimeter wave planar antennas. International Journal of High Speed Computing 10, 1–25 (1999)

7. Guiffaut, C., Mahdjoubi, K.: A parallel FDTD algorithm using the MPI library. IEEE Antennas and Propagation Magazine 43, 94–103 (2001)
8. Cummer, S.A.: A simple, nearly perfectly matched layer for general electromagnetic media. IEEE Microwave Wireless Compon. Lett. 13(3), 128–130 (2003)
9. Tirkas, P.A., Balanis, C.A., Renaut, R.A.: Higher order absorbing boundary conditions for the finite-difference time-domain method. IEEE Transaction on Antennas and Propagation 40, 1215–1222 (1992)

Application of Modified ICA to Secure Communications in Chaotic Systems

Shih-Lin Lin and Pi-Cheng Tung

The Department of Mechanical Engineering, the National Central University, Chung-Li 320, Taiwan
t331166@ncu.edu.tw

Abstract. Along with the increasingly quick development of modern communication system technologies, secure communication has become increasingly important. We propose a new method for secure communication systems. Independent component analysis (ICA) is employed to retrieve the message signal encrypted by a mixture of Gaussian white noise and chaotic noise. Unlike the traditional chaotic synchronization method, this method does not require knowing the basic Gaussian white noise and chaotic noise variables. However, the signals separated by traditional ICA shows opposite phase and unequal amplitude. Our study proposed a modified ICA, which can calculate accurately the phase and amplitude. The results showed that modified ICA could effectively extract the original message signal.

Keywords: Information Retrieval, Independent Component Analysis, Chaotic Systems, Secure Communication.

1 Introduction

Information security has become a major issue in recent years, as new ways of information exchange arise due to the rapid development of computing, communication, and internet technologies. This proposed secure communication system employed chaotic system for encryption as the broadband noise can effectively interfere with the original message signal. In 1963, Lorenz [1] developed the famous Lorenz equation, based on his weather experiments. In his computer simulation, he accidentally found the basic principle of chaos phenomenon: sensitive dependence on initial conditions. Since Lorenz, other scientists have developed many nonlinear equations, including Logistic Mapping, the Rössler system, the Duffing-Holmes system, and others, that have helped further the development of chaos theory. Generally speaking, a chaotic system includes the following characteristics:

1. Sensitive dependence on initial conditions: also known as the butterfly effect
2. Presence of the strange attractor: infinite messages within a finite space
3. Period-doubling phenomenon
4. Broadband observed in spectrum analysis

Due to the wide applicability of chaos theory in explaining all kinds of natural phenomenon, it has been employed in such diverse fields as biology, mathematics,

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, pp. 431 – 444, 2007.
© Springer-Verlag Berlin Heidelberg 2007

chemistry, physics, medicine, economics, geoscience, communications, and so forth. Many recent studies have offered the possibility of synchronizing chaotic systems and expressed its potential application for secure communications [2]-[23]. The two chaotic synchronization schemes most frequently studied are (1) conventional synchronization (CS) [2]-[5]and (2) generalized synchronization (GS) [3],[5],[6]. Murali et al [7] have discussed the use of encryption key functions for synchronization and communication through compound chaotic signal generation [7]. According to the reported studies, the three most common secure communication methods are: chaotic switching [4],[8]-[11], chaotic masking [4],[9],[12]-[15], and chaotic modulation [15]-[20]. Chaotic switching encodes two distinct chaotic attractors with bits "1" and "0" prior to detection by the receiver. Chaotic masking masks the signals before transmission. The message is added to the carrier which is also modulated by the information signal. This method was realized by Oppenheim et al. and Kocarev et al [21],[22]. Chaotic modulation encodes the transmitted signals with functions. The encoded signals are mixed with chaotic waves and transmitted from the master end. The slave end is composed of a chaos synchronization oscillator, which retrieves the signals by inverting the encodings. These traditional methods have their limits and cannot prevent distortion [2],[5],[7]-[25].

We propose the application of modified ICA for secure communication systems. ICA is a recently developed synthesizing technique that combines mathematics, physics, probability, statistics, computer simulation, neural network, and digital signal processing. It can extract the original signals from a mixture of measured signals from diverse sources. ICA can be used to analyze data originating from many different application fields. Makeig et al. analyzed electroencephalograms (EEG) with ICA to explain the relationship between finger movements and event-related potential (ERP) in human brain waves [26]. Stögbauer et al. proposed the application of ICA for the separation of maternal electrocardiographic (ECG) signals from fetal ECG signals, so as to improve the understanding of fetal conditions [27].

In this study we propose a new application of modified ICA, for secure communication. However, the signals separated by traditional ICA shows opposite phase and unequal amplitude, while the waveform of the estimated secured signals is the same. There are three defects in traditional ICA: 1) the amplitude does not equal the original signal; 2) there may be opposite phase; 3) the order will change. The former two have been indicated in many studies and are crucial to this consult [28]-[31]. This study proposed a modified ICA which will enable exact calculation of the phase and amplitude.

2 Proposed Secure Communications System

2.1 Secure Communications Scheme

The design of ICA, chaotic signals, and cryptography in secured communication is illustrated in Figure 1. At the transmitter, there are three signals. $s_1(t)$ indicates the message signals or original signals to be transmitted. $s_2(t)$ is the chaotic signal while $s_3(t)$ shows white Gaussian noise. The message signal is first encrypted with

traditional method. The encryption key will be sent to the receiver through a secure channel. The encrypted message will be further encrypted using chaotic signal and white Gaussian noise. The doubly-encrypted signal can be transmitted to the receiver end through public channel. At the receiver end, the message signal $s_1(t)$ can be recovered first by means of the modified ICA and then decrypted using the key received through the secure channel. The chaotic signal in the proposed system is comprised of uncorrelated white Gaussian noise. The $s_2(t)$ component is derived from the Lorenz attractor equation as follows [1]:

$$\begin{aligned}\frac{ds_2(t)}{dt} &= -\sigma s_2(t) + \sigma k(t)\\ \frac{dk(t)}{dt} &= -s_2(t)z(t) + rs_2(t) - k(t)\\ \frac{dz(t)}{dt} &= s_2(t)k(t) - bz(t)\end{aligned} \tag{1}$$

where σ, r and b are dimensionless parameters. The standard parameter values are $\sigma = 10, b = 8/3$ and $r = 28$. The equation can be solved by a 50 order Runge-kutta algorithm with a step size of 0.015. First, suppose the signal at the transmitter module is an 3-dimensional zero-mean vector, $\mathbf{s}(t) = [s_1(t), s_2(t), s_3(t)]^T$, and the components of s_i are independent. The vector $\mathbf{s}(t)$ is equivalent to 3 independent scalar-valued source signals $s_i(t)$. All the source signals must be statistically independent and have a non-Gaussian probability distribution. An exception is allowed for only one signal component.

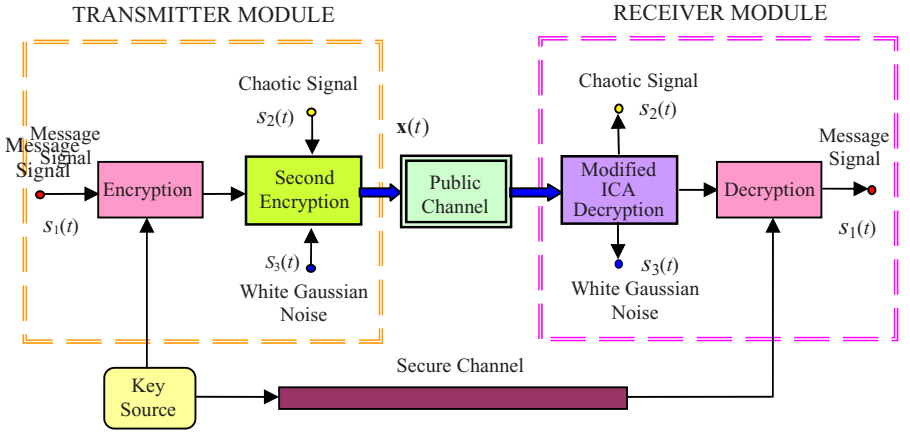


Fig. 1. Schematic diagram showing the secure communication scheme using the Modified ICA system

The $\mathbf{s}(t)$ signals enter an 3-by-3 $\mathbf{A}$ matrix. $\mathbf{A}$ is a full-rank matrix which can be called the mixing matrix. The $\mathbf{A}$ matrix makes the three signals interfere with each other, resulting in a mixed output, $\mathbf{x}(t)$. $\mathbf{x}(t)$ consists of three signals $x_1(t)$, $x_2(t)$, and $x_3(t)$. Their transmitter-module equations are as follows:

$$\begin{aligned} x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t) + a_{13}s_3(t) \\ x_2(t) &= a_{21}s_1(t) + a_{22}s_2(t) + a_{23}s_3(t) \\ x_3(t) &= a_{31}s_1(t) + a_{32}s_2(t) + a_{33}s_3(t) \end{aligned} \tag{2}$$

$a_{ij}\,(1 \le i, j \le 3)$ indicates the elements of $\mathbf{A}$ matrix. In this system, the a_{11}, a_{12}, and a_{13} must all be either one or at a known value, while other values can be an unknown and randomly selected value. $\mathbf{A}$ matrix will not be transmitted to the receiver. At this stage, the transmitted message signal $s_1(t)$ has been interfered by $s_2(t)$ and $s_3(t)$. This unrecognizable $s_1(t)$ fulfills the goal of secured communication. After encryption, he three signals $x_1(t)$, $x_2(t)$, and $x_3(t)$ are sent to the receiver through an public channel. Such this complicated encryption method can ensure the highest level of security in information exchange. When the receiver received $x_1(t)$, $x_2(t)$, and $x_3(t)$, the message signal can be recovered with ICA method. The estimation of $\mathbf{W}$ matrix is called the unmixing matrix, aiming at estimating $\mathbf{W} \approx \mathbf{A}^{-1}$ .The equation derived after estimating $\mathbf{W}$ may be shown as follows:

$$\begin{aligned} u_1(t) &= w_{11}x_1(t) + w_{12}x_2(t) + w_{13}x_3(t) \\ u_2(t) &= w_{21}x_1(t) + w_{22}x_2(t) + w_{23}x_3(t) \\ u_3(t) &= w_{31}x_1(t) + w_{32}x_2(t) + w_{33}x_3(t) \end{aligned} \tag{3}$$

Here w_{ij} ($1 \le i, j \le 3$) is an element of $\mathbf{W}$ matrix. $\mathbf{u}(t) = [u_1(t), u_2(t), u_3(t)]^T$ is the estimated message signal.

An informax ICA can identify $\mathbf{W}$ and retrieve the message signal. Statistical independence can be measured by entropy. Entropy is the basic concept of information theory [32]. It indicates the degree of indeterminacy of random variables. In other words, the more unpredictable and unstable a variable is, the greater the entropy and the greater its statistical independence. The following equation defines the entropy H of a binomial random variable. It can be extended to continuous random variable and random vectors. While the random vector is y, the density distribution is $p(y)$:

$$H(\mathbf{y}) = -\int p(y) \log p(y) dy \tag{4}$$

After measuring the statistical entropy, the greatest entropy may be measured by mutual information, as shown in the following equation:

$$I(y_1, y_2, y_3) = H(y_1) + H(y_2) + H(y_3) - H(y_1, y_2, y_3). \tag{5}$$

When the output entropy $H(y_1, y_2, y_3)$ is at its greatest value, the mutual information $I(y_1, y_2, y_3)$ between the outputs is the smallest. When $I(y_1, y_2, y_3) = 0$, y_1, y_2 and y_3 are statistically independent. The relation between mutual information and the $\mathbf{W}$ value, uncovered by the scholarly efforts from related fields, may be represented as the following

$$\Delta \mathbf{W} \propto \frac{\partial H(y)}{\partial \mathbf{W}} = (\mathbf{W}^T)^{-1} + \begin{pmatrix} \frac{\partial p(\mathbf{u})}{\partial \mathbf{u}} \\ \hline p(\mathbf{u}) \end{pmatrix} \mathbf{x}^T. \tag{6}$$

$\Delta \mathbf{W}$ is defined as the modified $\mathbf{W}$, $\mathbf{u} = [u_1, u_2, u_3]$, and $p(\mathbf{u}) = \left| \frac{\partial y}{\partial \mathbf{u}} \right|$. However, the learning rule is too complicated as it involves the operation of inverse matrix. In the studies of Amari et al. in 1996 [33], the learning rule was multiplied by $\mathbf{W}^T\mathbf{W}$,resulting in its rescale. In consequence, the learning rule is changed into

$$\Delta \mathbf{W} \propto \frac{\partial H(y)}{\partial \mathbf{W}} \mathbf{W}^T \mathbf{W} = \left[I + \varphi(\mathbf{u}) \mathbf{u}^T \right] \mathbf{W}. \tag{7}$$

$\varphi(\mathbf{u})$ is defined as the $\begin{pmatrix} \frac{\partial p(\mathbf{u})}{\partial \mathbf{u}} \\ \hline p(\mathbf{u}) \end{pmatrix}$. The maximum information method is to use estimate ascent algorithm, adjusting $\mathbf{W}$ by means of continuous iteration and achieving the greatest $H(\mathbf{y})$. The adjustment is presented as such

$$\mathbf{W}_{p+1} = \mathbf{W}_p + l \Delta \mathbf{W} \tag{8}$$

where p is the times of iteration and l stands for the learning rate. $\mathbf{W}$ will be renewed continuously according to the partial differential equation of the maximized function $H(\mathbf{y})$ in relation to $\mathbf{W}$. $H(\mathbf{y})$ will increase until the greatest value is identified. So the best $\mathbf{W}$ may be estimated, enabling the separation of original signals from the mixed signals. As the $\mathbf{W}$ estimated by ICA is not the actual $\mathbf{A}$, so is $\mathbf{u}(t)$ not the actual $\mathbf{s}(t)$. However, they also show the problems of unequal amplitude and opposite phase as in waveforms.

2.2 Proposed Modified ICA

Our study proposed a modified ICA, which can calculate accurately the phase and amplitude. As shown in Fig. 2, gradient method is employed to calculate the phase and amplitude of the original signal. After recovering $u_1(t)$, $u_2(t)$, $u_3(t)$ with ICA, the hypothesized gains $\theta_1(t)$, $\theta_2(t)$, and $\theta_3(t)$ in gradient estimate can be sued to solve the

problem. The positive and negative values of the gains can adjust positive or opposite phase while the amount of gains can be used in dealing with unequal amplitude. In the gradient method, two inputs are required. One is $x_1(t)$ the original mixed signal. The other is $\mathbf{u}(t)$ of the gains produced from ICA and calculated by gradient method. The gain will be updated continuously until the two signals have the same value. At this point, the system converges and the best gain is identified. The phase and amplitude of the original signal can be calculated by ICA input multiplied the new gains. As shown previously, the modified ICA aims at estimating the actual phase. The limited a_{11}, a_{12}, and a_{13} are special cases. If the value is 1 or a_{11}, a_{12}, and a_{13} are known,

$$s_1(t) \approx \theta_1(t) u_1(t) \tag{9}$$

$$s_2(t) \approx \theta_2(t) u_2(t) \tag{10}$$

$$s_3(t) \approx \theta_3(t) u_3(t) . \tag{11}$$

The gains of θ_1, θ_2, and θ_3 can be automatically adjusted. The modified algorithm can be written as

$$d(t) = \theta_1(t) u_1(t) + \theta_2(t) u_2(t) + \theta_3(t) u_3(t) \tag{12}$$

where $d(t)$ is modified algorithm output, $u_1(t)$ is estimation of message signal, $u_2(t)$ indicates chaotic signal and $u_3(t)$ represents white Gaussian noise. $\theta_1(t)$, $\theta_2(t)$ and $\theta_3(t)$ are gain vector at time t. Estimation error or error signal represents the

$$e(t) = x_1(t) - d(t) . \tag{13}$$

Thus, substituting (12) into (13), we may compute the updated value of the gain vector $\theta(t+1)$ by using the simple recursive relation

$$\theta_1(t+1) = \theta_1(t) + \mu [e(t) u_1(t)] \tag{14}$$

$$\theta_2(t+1) = \theta_2(t) + \mu [e(t) u_2(t)] \tag{15}$$

and $$\theta_3(t+1) = \theta_3(t) + \mu [e(t) u_3(t)] . \tag{16}$$

In this gradient algorithm, the parameter μ is the step size, which mainly affects the stability and convergence speed of the system. When $e(t)$ is 0, three constants $\theta_1(t)$, $\theta_2(t)$ and $\theta_3(t)$ are derived. The actual message signal can therefore be recovered. Fig. 3 shows Signal-flow graph of the modified ICA.

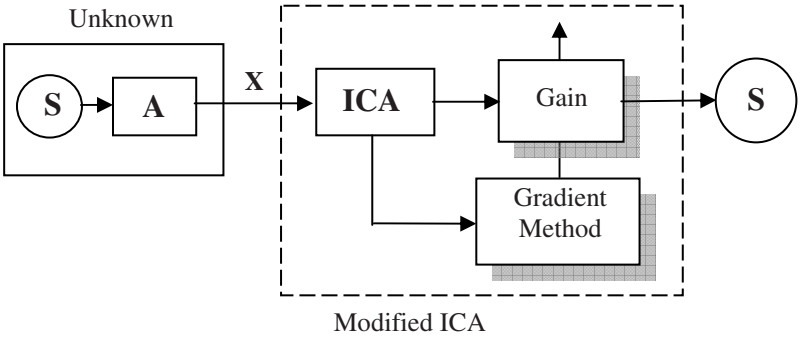


Fig. 2. The structure of the modified ICA

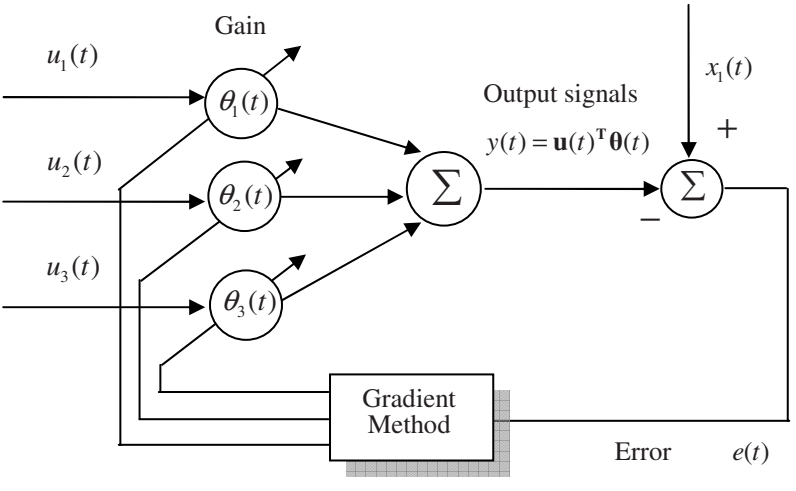


Fig. 3. Signal-flow graph of the modified ICA

3 Experiment

This study uses wireless communication technology to transmit secured signal from the transmitter to the receiver. As shown in Figure 4, the Wireless Communication Model RS-232 is capable of transmitting and receiving messages. At the work frequency of at 433-435 MHz. Modulation is controlled by FSK system. The transmission power is 12dBm, with receiver sensitivity at -112dBm, operation voltage at DC 6~10 V, communication distance at 200-600m (which varies by environment), and receiving current at 35mA. The type connector is RS232 D-SUB 9 pins. The operation temperature is -5℃ ~ 60℃. The red LED is the transmitter end and the green one is the receiver end. Table 1 Shows hardware specifications of the communication device.

Connect this model to the RS-232 on two computers for wireless data transmission. One computer transmits the simulated signal. The other receives the signal, conducting decryption and decoding to the specificity of the seventh digit after the decimal point. No aliasing is allowed in the transmission and reception of the signal.

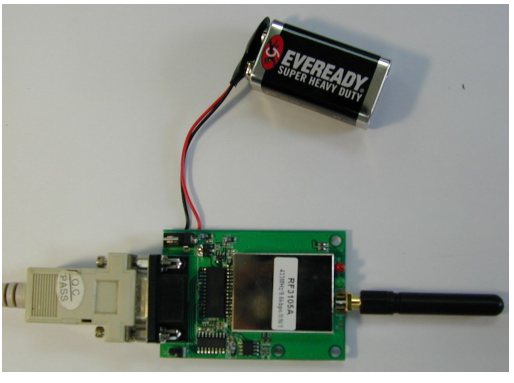


Fig. 4. Wireless communication model

Table 1. Hardware specifications of the communication device

Working Frequency	433-435 MHz
Modulation	FSK
Channels	At least 25 channels
Functions	Half-Duplex, Bi-direction, Transparent wireless data transmit and receive.
Transmission Power	12dBm
Receiver Sensitivity	-112dBm@4.8 kbps
Operation Voltage	DC 6~10 V
Baud Rates	1.2/ 4.8/ 9.6/ 19.2/ 38.4/ 76.8 Kbps
Communication Distance	200-600m (Whip antenna, outdoor, depending on different communication speeds)
Transmission Current	85mA@12dBm
Receiving Current	35mA
Type Connector	RS232 D-SUB 9 pins
Operation Temperature	-5℃ ~ 60℃
LED	Red: Transmitting; Green: Receiving

4 Results and Discussion

Secure communication in chaotic systems is aimed at protecting the message signal by the use of chaotic signals and white Gaussian noise. With mixed signals, other people will not be able to retrieve the useful message. Only the intended receiver can retrieve the message signal. Thus the transmission of important message can be duly secured. In this study, we use a CPU-P4 2.0 GHz industrial computer with the

MATLAB 7.0 software. Fig. 5 indicates the chaotic signal of the Lorenz attractor used. It can be described by (1). The parameters were explained earlier. Here $s_2(t)$ designates the chaotic signal and the 135 Hz sin signal is the message signal. A normally distributed random number with a mean of 0 and a variance of 225 indicates the broad-band white Gaussian noise. Fig. 6 shows time and frequency domain of the three signals. To adapt to an austere environment, the amplitude of the message signal is the smallest, whereas those of the chaotic signal and of white Gaussian noise are greater. Here the message signal is a 135Hz narrow-band signal. The frequency domain of the chaotic signal is within 400 Hz, and that of the white Gaussian noise is within 4000 Hz. The three signals were processed for second encryption as in (2). Second encryption is realized by allowing $s_1(t)$ 、 $s_2(t)$ and $s_3(t)$ to pass through the 3-by-3 nonsingular mixing matrix $\mathbf{A}$. This will result in mutual interference between the signals, as shown in Fig. 7. Here the original message signal cannot be distinguished from the other signals. The original message signal is concealed within the mixture of the chaotic signal and the white Gaussian noise, and is therefore secured. For transmission, the three signals must be transmitted simultaneously through different channels. Fig. 8 shows time and frequency domain of three mixed signals from receiver. Upon reception, the signals can be retrieved by estimating $\mathbf{W} \approx \mathbf{A}^{-1}$ with (3), (5), (6), and (7), as presented in Fig. 9. Fig. 9 shows the signal retrieved by traditional ICA. Opposite phase and unequal amplitude can be observed. Fig. 10 presents the signal retrieved by modified ICA. The step-size μ is 0.01. The phase and amplitude of the original signal can be duly calculated.

There are several conditions and constraints for the application of ICA to secure communications in chaotic systems. All the independent components s_i (i.e., the message signal, the chaotic signal, and the white Gaussian noise) must be statistically independent, which means they must have the relation $p(s_1, s_2..., s_i) = p_1(s_1)p_2(s_2)...p_n(s_i)$. All the independent components s_i must be non-Gaussian in distribution. Based on the previous assumption, the application of ICA on the system may be scribed. $s_1(t)$, $s_2(t)$, and $s_3(t)$ are statistically independent. As the signal goes through mixing matrix, it becomes correlated and therefore statistically dependent, as in (2). If we can recover the signal to its original statistical independence, this means that the signal is successfully separated. Hence, two questions arise. First, how to determine whether the signal is statistically independent? Second, how to identify the correct $\mathbf{W}$? For the first question, negentropy is proposed to measure non-Gaussianity. If we can measure non-Gaussianity, we can determine its independence. The second question can be explained by minimization of mutual information. It explains the relationship between $\mathbf{W}$ and negentropy. In (4), we can use negentropy to measure the independence of the output $\mathbf{u}(t)$, adjust $\mathbf{W}$ to make it approach independence, and use optimal algorithm to find the best $\mathbf{W}$.

This is crucial for the efficacy of retrieving signals by ICA. However, according to the central limit theorem, one component is allowed to have a Gaussian distribution. The number of the received mixed signal $\mathbf{x}(t)$ must be greater or equal to that of the independent components. The proposed system is structured differently than the

traditional chaotic communication systems. Unlike a traditional system that transmits only one signal, this system requires the transmission of multiple signals in order to retrieve the message.

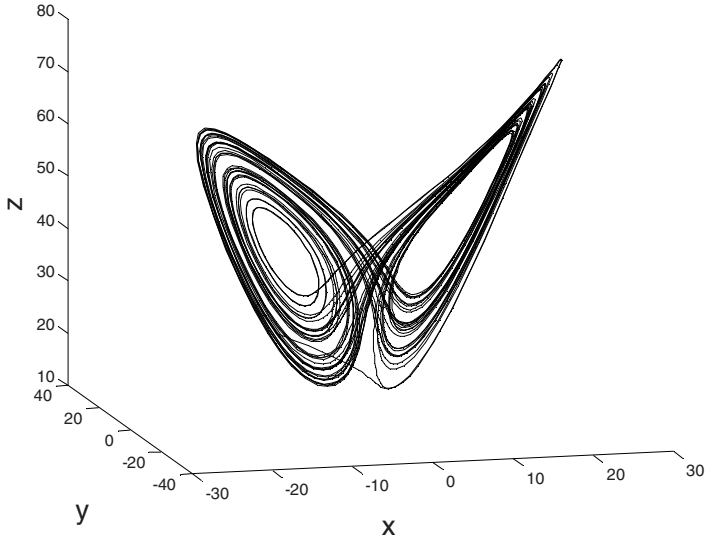


Fig. 5. Basic Lorenz attractors of a chaotic system

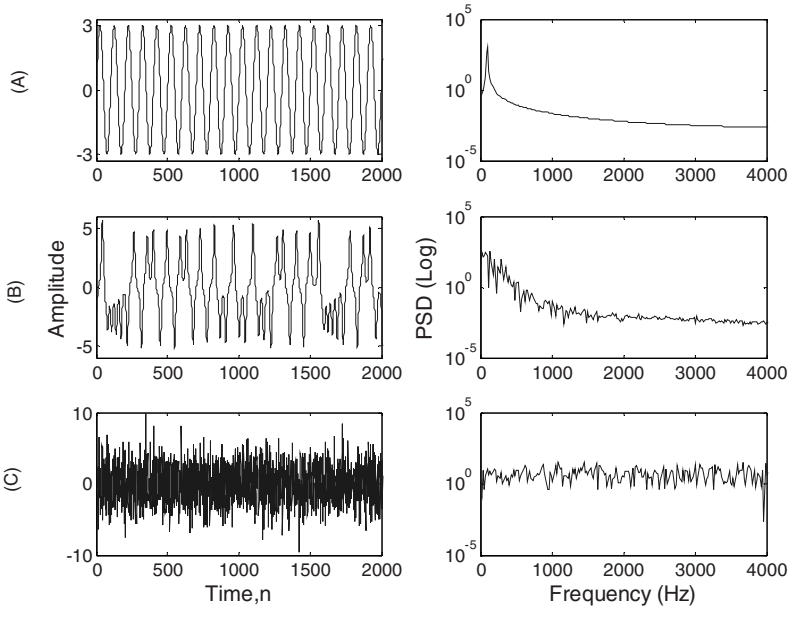


Fig. 6. Time and frequency domain of the three signals: (A) Message signal (B) Chaotic signal (C) White Gaussian noise

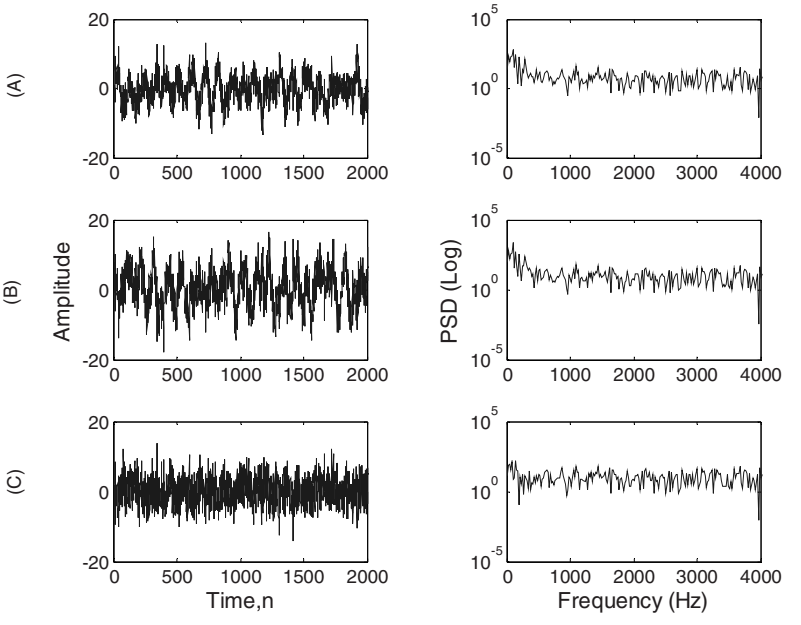


Fig. 7. Time and frequency domain of three mixed signals from transmitter

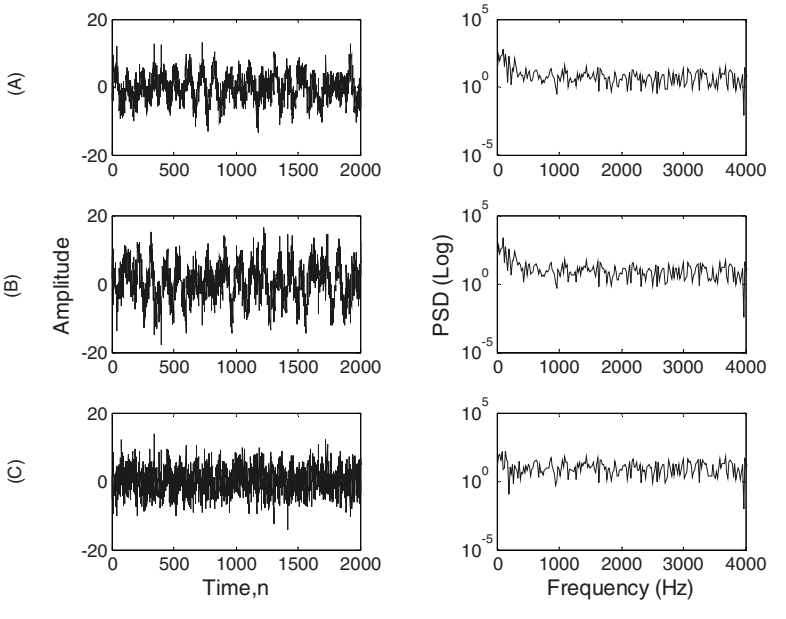


Fig. 8. Time and frequency domain of three mixed signals from receiver

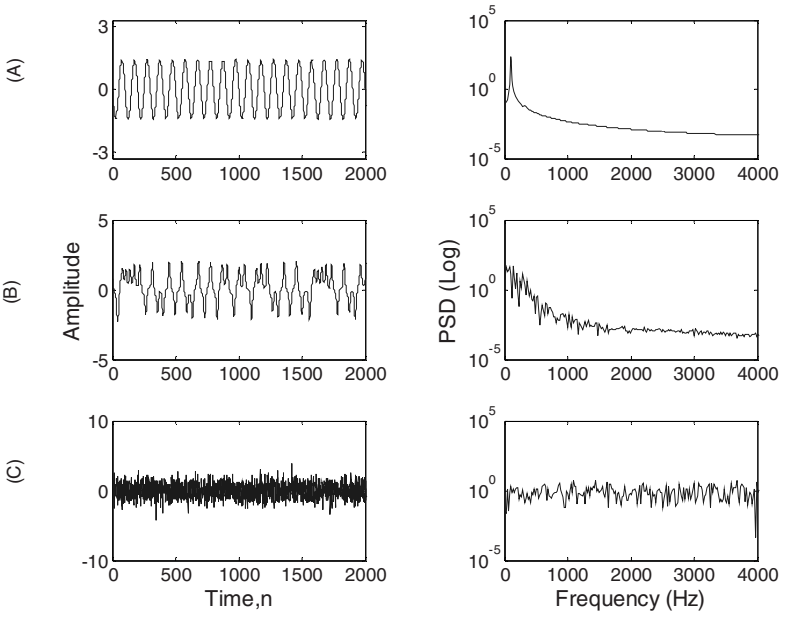


Fig. 9. Time and frequency domain of the signals separated by traditional ICA: (A) Estimation of message signal (B) Estimation of chaotic signal (C) Estimation of white Gaussian noise

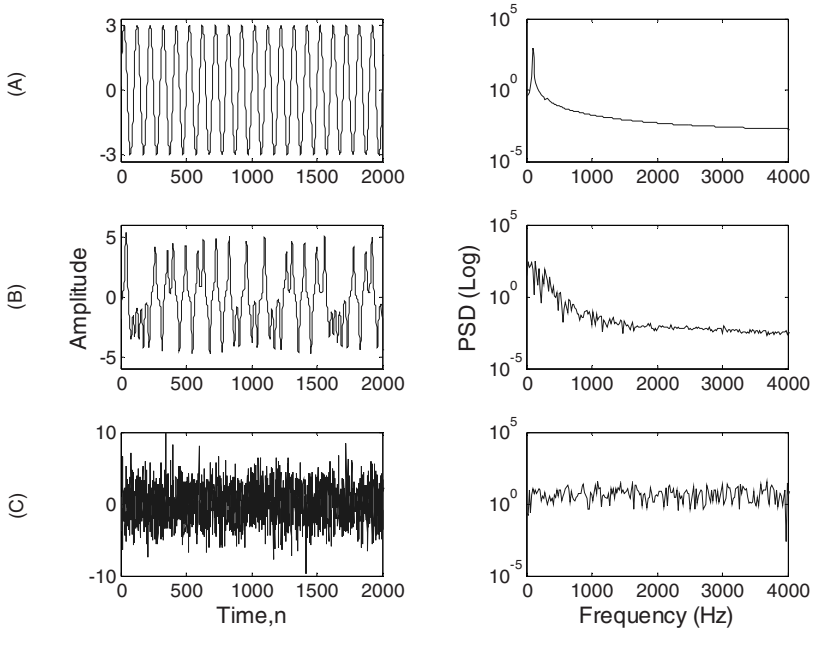


Fig. 10. Time and frequency domain of the signals separated by modified ICA: (A) Estimation of message signal (B) Estimation of chaotic signal (C) Estimation of white Gaussian noise

5 Conclusion

This paper proposes a new method for secure communication systems. Modified ICA was employed to retrieve the message signal encrypted by its mixture with white Gaussian noise and chaotic noise. The results show that the method can retrieve very weak message signals effectively. The retrieved signal carries the same characteristics as the original message signal. The proposed method provided the following advantages: low amount of calculation, simplicity, fast convergence speed, and excellent security. Unlike the traditional chaotic synchronization, it does not require knowing the basic variables of chaotic noise and white Gaussian noise. However, there are several constraints in the application of modified ICA to secure communications in chaotic systems. All the independent components must be statistically independent. The number of mixed signals received must be larger than or equal to the number of independent components. Using modified ICA as a tool, a mixture of message signal, chaotic signal and white Gaussian noise can be separated and retrieved with successful results. Modified ICA has a great potential for application in secure communications.

Acknowledgments. This project was supported by the National Science Council in Taiwan, Republic of China, under Project Number NSC 95-2218-E-008-005.

References

1. Lorenz, E.N.: Deterministic Nonperiodic Flow. J. Atmos. Sci. 20, 130–141 (1963)
2. Carroll, T.L.: Communicating with use of Filtered Synchronized Chaotic Signals. IEEE Trans. Circuits Syst. 42, 105–110 (1995)
3. Rulkov, N.F., Sushchik, M.M., Tsimring, L.S., Abarbanel, H.D.I.: Generalized Synchronization of Chaos in Directionally Coupled Chaotic Systems. Phys. Rev. E 51, 980–994 (1995)
4. Lakshmanan, M., Murali, K.: Chaos in Nonlinear Oscillators: Controlling and Synchronization. World Scientific, Singapore (1996)
5. Kocarev, L.j., Parlitz, U.: Generalized Synchronization, Predictability, and Equivalence of Unidirectionally Coupled Dynamical Systems. Phys. Rev. Lett. 76, 1816–1819 (1996)
6. Murali, K., Lakshmanan, M.: Secure Communication using a Compound Signal from Generalized Synchronizable Chaotic Systems. Phys. Lett. A 241, 303 (1998)
7. Murali, K.: Digital Signal Transmission with Cascaded Heterogeneous Chaotic System. Phys. Rev. E 63, 16217 (2000)
8. Parlitz, U., Chua, L.O., Kocarev, L.j., Halle, K.S., Shang, A.: Transmission of Digital Signals by Chaotic Synchronization. Int. J. Bifurcation Chaos 2(4), 973–977 (1992)
9. Cuomo, K.M., Oppenheim, A.V.: Circuit Implementation of Synchronized Chaos with Applications to Communications. Phys. Rev. Lett. 71, 65–68 (1993)
10. Dedieu, H., Kennedy, M.P., Hasler, M.: Chaos Shift Keying: Modulation and Demodulation of a Chaotic Carrier using Self-synchronizing Chua's Circuits. IEEE Trans. Circuits Syst. 40, 634–642 (1993)
11. Parlitz, U., Ergezinger, S.: Robust Communication Based on Chaotic Spreading Sequences. Phys. Lett. A 188(2), 146–150 (1997)
12. Kocarev, L.j., Halle, K., Eckert, K., Chua, L.O., Parlitz, U.: Experimental Demonstration of Secure Communications via Chaotic Synchronization. Int. J. Bifurcation Chaos Appl. Sci. Eng. 2(2), 709–713 (1992)

13. Murali, K., Lakshmanan, M.: Transmission of Signals by Synchronization in a Chaotic Van der Pol–Duffing Oscillator. Phys. Rev. E 48, R1624-R1626 (1993)
14. Wu, C.W., Chua, L.O.: A Simple way to Synchronize Chaotic Systems with Applications to Secure Communication Systems. Int. J. Bifurcation Chaos Appl. Sci. Eng. 3, 1619–1627 (1993)
15. Ogorzalek, M.J.: Taming Chaos. I. Synchronization. IEEE Trans. Circuits Syst. 40, 639–699 (1993)
16. Halle, K.S., Wul, C.W., Itoh, M., Chua, L.O.: Spread Spectrum Communication Through Modulation of Chaos. Int. J. Bifurcation Chaos Appl. Sci. Eng. 3, 469–477 (1993)
17. Dmitriev, A.S., Panas, A.I., Starkov, S.O.: Experiments on Speech and Music Signals Transmission using Chaos. Int. J. Bifurcation Chaos Appl. Sci. Eng. 5(4), 1249–1254 (1995)
18. Dmitriev, A.S., Panas, A.I., Starkov, S.O., Kuzmin, L.V.: Experiments on RF Band Communications using Chaos. Int. J. Bifurcation Chaos Appl. Sci. Eng. 7, 2511–2527 (1997)
19. Murali, K., Lakshmanan, M.: Efficient Signal Transmission by Synchronization Through Compound Chaotic Signal. Phys. Rev. E 56, 251–255 (1997)
20. Murali, K., Lakshmanan, M.: Secure Communication using a Compound Signal from Generalized Synchronizable Chaotic Systems. Phys. Lett. A 241, 303–310 (1998)
21. Oppenheim, A.V., Wornell, G.W., Isabelle, S.H., Cuomo, K.M.: Signal Processing in the Context of Chaotic Signals. IEEE Trans., 117–120 (1992)
22. Kocarev, L.J., et al.: Synchronization in Chaotic Systems. Int. J. of Bifur Chaos 2, 709–713 (1992)
23. SJ, L., Alvarez, G., Chen, G.R.: Breaking a Chaos-Based Secure Communication Scheme Designed by an Improved Modulation Method. Chaos Solitons & Fractals 25(1), 109–120 (2005)
24. Zhou, J., Huang, H.B., Qi, G.X., et al.: Communication with Spatial Periodic Chaos synchronization. Physics Lett. A 335(2-3), 191–196 (2005)
25. Kim, C.M., Kye, W.H., Rim, S., et al.: Communication key using delay times in time-Delayed Chaos Synchronization. Physics Lett. A 333(3-4), 235–240 (2004)
26. Makeig, S., Westerfield, M., Jung, T-P., Enghoff, S., Townsend, J., Courchesne, E., Sejnowski, T.J.: Dynamic Brain Sources of Visual Evoked Responses. Science 295, 690–694 (2002)
27. Stögbauer, H., Kraskov, A., Astakhov, S.A., Grassberger, P.: Least Dependent Component Analysis Based on Mutual Information. Phys. Rev. E 70, 66123 (2004)
28. Lauro, E.D., Martino, S.D., Falanga, M., Ciaramella, A., Tagliaferri, R.: Complexity of Time Series Associated to Dynamical Systems Inferred from Independent Component Analysis. Physical Review E 72, 46712 (2005)
29. Khor, L.C., Woo, W.L., Dlay, S.S.: Nonlinear Blind Signal Separation with Intelligent Controlled Learning. IEE Proc.-Vis. Image Signal Process. 152(3), 297–306 (2005)
30. Fiori, S.: Fixed-Point Neural Independent Component Analysis Algorithms on the Orthogonal Group. Future Generation Computer Systems 22, 430–440 (2006)
31. Gonzalez de la Rosa a, J.J., Puntonet b, C.G., Lloret, I.: An Application of the Independent Component Analysis to Monitor Acoustic Emission Signals Generated by Termite Activity in Wood. Measurement 37, 63–76 (2005)
32. Bell, A.J., Sejnowski, T.J.: An Information Maximisation Approach to Blind Separation and Blind Deconvolution. Neural Comput. 7(6), 1129–1159 (1995)
33. Amari, S., Cichocki, A., Yang, H.H.: A New Learning Algorithm for Blind Signal Separation. Advances in Neural Information Processing Systems 8, 757–763 (1996)

Zero Memory Information Sources Approximating to Video Watermarking Attacks

M. Mitrea[1], O. Dumitru[1], F. Prêteux[1], and A. Vlad[2,3]

[1] ARTEMIS Department, National Institute on Telecommunications, Evry-France
[2] Faculty of Electronics, Telecommunications and Information Technology, POLITEHNICA University of Bucharest-Romania
[3] The Research Institute for Artificial Intelligence, Romanian Academy
{mihai.mitrea, octavian.dumitru, francoise.preteux}@int-edu.eu, adriana_vlad@yahoo.com

Abstract. Under the framework of the watermarking applications, to accurately model the malicious attacks becomes nowadays a challenging research topic. Following a previous study pointing out the inner limitations of the popular Gaussian model, this paper computes for the first time accurate approximations for the zero memory information sources describing some main real life attacks in video watermarking: linear and non-linear filtering, rotations, and StirMark (which may emulate the in-theatre camcorder capture). In order to obtain these models, the authors advance an original and generic statistical approach for *pdf* estimation combining EM Gaussian mixtures and confidence limits. This investigation procedure does not rely on any *a priori* assumption concerning the video/attack stationarity and handles with mathematical rigour the dependency existing among successive frames in a video sequence.

Keywords: watermarking attack, *pdf* estimation, stationarity.

1 Introduction

The emerging IT society, founded on ambient intelligence, ubiquitous computing and fixed/mobile convergence puts art into a completely new perspective: the production, dissemination and consumption of any type of digital content is just a click/scroll away from any user. However, the art producers are still very reluctant. The main concern is related to property right protection in a world where the copying is perfect, instantaneous and with virtually no cost. For instance, the last IIPA (International Intellectual Property Alliance) report states that the DVD piracy is up to 3 time more profitable than the old-fashioned heroine traffic [1].

To address such an issue, the scientific community proposes two families of solutions. The first one is based on DRM/IPMP (Digital Right Management / Intellectual Property Management and Protection) and tries to impose proprietary hardware/software restrictions in player and/or copier devices [2]. Such a solution, fostered by large companies, is seen in many countries (EU, USA, …) as a violation of the user fundamental rights (he/she may be forced to chose a devoted device for a particular type of content [2], thus resulting in losses of about 40% in sales [3]). The

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 445 – 459, 2007.
© Springer-Verlag Berlin Heidelberg 2007

second family of solutions is represented by watermarking and aims at persistently and imperceptibly associating some copyright information with any multimedia content. In contrast to DRM/IPMP, watermarking does not impose any constraints to a legal user but allows the illegal copies to be spotted out and the copy-maker to be tracked-down. Of course, hybrid DRM/watermarking solutions can be considered.

Let us now present the watermarking main definitions and theoretical background. In its largest acceptation, the term *robust watermarking* designates any method that imperceptibly inserts some data - referred to as *mark* or *watermark* - into a host document [4-5], be it video, audio, 3D, …; it should not be possible for these additional data to be removed by normal or malicious processing. The watermark is generated starting from some public information (*e.g.* the copyright information, a logo, …) and from some secret information - the *key* - available only for the copyright holder.

For a real life copyright application, a watermarking technique should feature good *transparency*, high *robustness*, a quite large *data payload*, very low *probability of false alarm*, and *obliviousness*.

- The *transparency* refers to the imperceptibility of the watermark in the document. That is, a human observer should not perceive any difference between the marked and un-marked contents.
- The *robustness* (association persistency) is the ability of the watermark to survive any processing that may be applied to the marked data: mundane processing (compression, resizing, …) and malicious *attacks* (*i.e.* transformations applied to the marked content with the stated purpose of removing the watermark). In the present paper, by attacks we shall refer to both malicious and mundane transformations.
- The *data payload* is the quantity of information that is inserted into the host document. For a basic copyright application, the data payload may be relatively small (*e.g.* 64 bits allow a mobile user to be identified by his/her IMSI - International Mobile Subscriber Identity); however, for advanced services, like hybrid DRM/watermarking or *in-band enriched multimedia* [6], a much larger data payload would be required.
- The *probability of false alarm* is the probability of taking an unmarked document for a marked one. For a practical application, such a probability should be lower than a certain upper limit, *e.g.* $P_f < 10^{-10}$.
- The *obliviousness* refers to the possibility of detecting the mark in the absence of the original document.

From the theoretical point of view, a generic watermarking system can be modelled as a noisy channel, Fig. 1. The mark is not generally directly inserted into the video sequence but rather into a certain transform of it (like the DCT – Discrete Cosine Transform or DWT – Discrete Wavelet Transform), denoted by T in Fig. 1. The mark is a sample from the information source and should be recovered at the detection side (*i.e.* it should be detected in the marked document). The elements that make the watermark detection difficult can be modelled as the channel noise, namely: the original content and the attacks. The watermarking procedure itself plays the role of the modulation technique (*i.e.* the way in which the mark is inserted into the host content). This model brings into evidence that the design of a reliable watermarking

technique cannot be achieved without an in-depth knowledge of the noise (*i.e.* original content and attacks) statistical model. Despite suspicion and reticence, in the absence of any sound statistical investigation, such a model is generally assumed to be Gaussian. In their previous works [7-9], the authors identified the validity perimeter of this assumption and pointed out to the Gaussian model inner limitations.

The present paper goes deeper into this study and provides for the first time accurate statistical models for some real life attacks (Gaussian filtering, sharpening, rotations, StirMark), when the mark is inserted in the DWT coefficient hierarchy [10].

The paper is structured according to the main stages which are compulsory when advancing any model: definition (Section 2), computation (Section 3) and validation (Section 4). Section 5 concludes the paper and opens the perspective for future works.

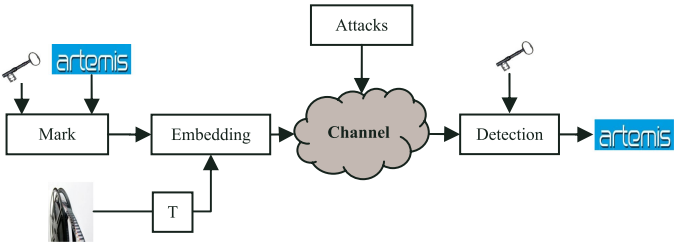


Fig. 1. Watermarking as a noisy channel: the signal to be transmitted is the mark (generated from a logo, according to a secret key); the noise is represented by the T transform of the original video and by the attacks

2 Model Definition

2.1 Watermarking Attacks as Information Sources

This sub-section outlines the way in which we represent the watermarking attacks by information sources [9].

Algorithm 1

Be there a colour video sequence consisting in L frames. Each frame is represented in the HSV (Hue-Saturation-Value) space; the V component is normalised to the $[0,1]$ interval. The following 7 steps should be gone through for each frame in the video sequence, Fig. 2:

Step 1: Apply the 2D-DWT to the V (luminance component), at an N_r resolution level.

Step 2: Record in a vector the DWT coefficients corresponding to the LH and HL lowest frequency sub-bands.

Step 3: Build up the coefficient hierarchy by sorting in a decreasing order the vector obtained in the previous step, and record the largest R ranks alongside with their spatial locations; the resulting vectors are denoted by *original_coefficients* and *original_locations*, respectively.

Step 4: Apply the considered attack to the processed frame.
Step 5: Resume the Step 1 on the attacked video sequence.
Step 6: Record in a vector denoted by *attacked_coefficients* the coefficients in the attacked frames corresponding to the *original_locations*.
Step 7: Compute the difference between the two vectors:

difference =attacked_coefficients - original_coefficients.

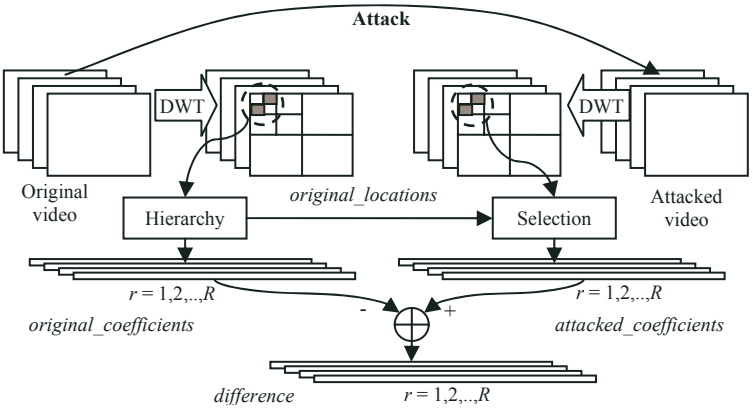


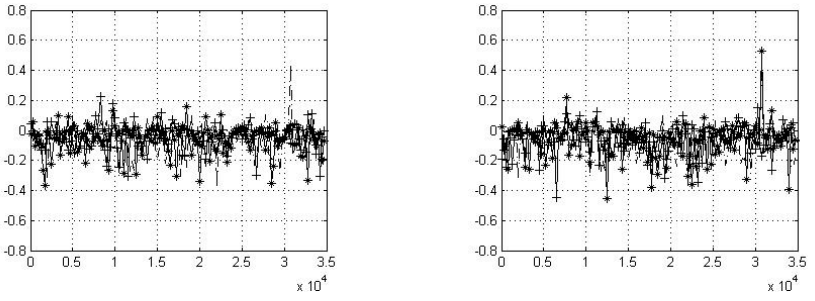
Fig. 2. Obtaining a sample from the information sources modelling the watermarking attacks

Comments concerning Algorithm 1

After performing the *Algorithm 1* (*i.e.* after applying the Steps 1-7 to each frame in the video sequence), a set of L vectors (with R components) of the same type as *difference* is obtained. Should we consider only the values corresponding to an arbitrarily chosen rank r, a new vector of L components is obtained: $noise = [n_1, n_2, ..., n_L]$. An n_j component ($j = 1,2,..., L$) in a *noise* vector is the value corresponding to the rank r in the *difference* vector computed on the frame j. The *noise* vector components are random: they depend at least on the original frame content and on the random mechanism (if any) in the attack. From the information theory point of view, we may consider that *each rank* is associated to a continuous information source. The challenging issue of this study is to answer for the first time the following two questions:

1. Does such an information source feature stationarity? In other words, is there a general model for an attack or does it depend on the investigated video sequence (Fig. 3.a) and/or on the time origin (Fig. 3.b)?
2. Which is the zero memory approximation to this information source? In other words, which is the *pdf* (*p*robability *d*ensity *f*unction) for the random variable approximating this source?

Note that our previous study [11] brought into evidence that for the large majority of the attacks and for practically all the investigated ranks, the Gaussian *pdf* cannot be



(a) (b)

Fig. 3. Samples from the information sources approximating the attacks (L=35000), for an arbitrarily chosen rank: the three plots in (a) correspond to three different video sequences while the 3 plots in (b) to the same video sequence but to three different time origins

accepted. As that investigation was based on four types of statistical tests (Chi-square, Ro, Student and Fisher), now we shall approach the modelling problem by means of estimation theory.

2.2 *Pdf* Estimation Tools

This sub-section presents two different basic pdf estimation tools: EM (expectation-maximisation) Gaussian mixtures and confidence limits. The former method will be involved in model specification (Section 2.3) while the latter will be considered as a cross-checking procedure during the model validation (Section 4).

Be there $[x_1, x_2, ..., x_N]$ a sample of N experimental data complying with the *iid* model (*i*ndependent and *i*dentically *d*istributed data). We suppose that these data are sampled from a X random variable whose $p(x)$ *pdf* is unknown and should be estimated.

The *Gaussian mixtures* (see [12] and the references therein) are linear combinations of Gaussian laws which can approximate any continuous *pdf*. The $\hat{p}(x)$ estimate of the $p(x)$ function can be computed by (1):

$$\hat{p}(x) = \sum_{k=1}^{K} P(k) p_k(x), \text{ where } p_k(x) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right). \quad (1)$$

In (1), the number of mixtures K is pre-established by the experimenter and $P(k), \mu_k$ and σ_k are $3K$ parameters to be estimated on the data sample (they refer to the mean values, to the standard deviations and to the weights of the considered Gaussian laws in the mixture). In this respect, an EM (expectation maximisation) algorithm (considering a maximum likelihood based criterion) is performed by (2):

$$\text{The } E \text{ step: } p^{(i)}(k/x_n) = \frac{p_k^{(i)}(x_n) P^{(i)}(k)}{\hat{p}^{(i)}(x_n)}, \quad (2.a)$$

$$\text{The } M \text{ step: } \mu_k^{(i+1)} = \frac{\sum_{n=1}^{N} p^{(i)}(k/x_n) x_n}{\sum_{n=1}^{N} p^{(i)}(k/x_n)},$$

$$(\sigma_k^2)^{(i+1)} = \frac{\sum_{n=1}^{N} p^{(i)}(k/x_n)\left(x_n - \mu_k^{(i+1)}\right)^2}{\sum_{n=1}^{N} p^{(i)}(k/x_n)}, P^{(i+1)}(k) = \frac{1}{N} \cdot \sum_{n=1}^{N} p^{(i)}(k/x_n), \quad (2.b)$$

where the (i) upper index denotes the current iteration; the total number of iterations is also subject to the experimenter choice. Concerning the initialisation step, we adopted a random rule matched to the experimental data.

As it can be seen, this procedure estimates the searched *pdf* by an analytic function whose parameters are computed on the experimental data. The *pdf* estimation with confidence limits follows a different philosophy. It starts from the idea that the $p(x)$ *pdf* to be estimated is generally employed to compute the probability that the X investigated random variable takes values in a given I interval. Consequently, it would be useful to directly estimate this probability and not the *pdf per-se*; in other words, *pdf* estimation with confidence limits means to compute confidence limits for $\int_I p(x)dx$ and not for $p(x)$, as follows.

Be there $x_{\min}$ and $x_{\max}$ the extreme values in the $[x_1, x_2, ..., x_N]$ sample. We consider a division of the $[x_{\min}; x_{\max}]$ interval into a set of S sub-intervals denoted by I_s, $s = 1,2,...S$. Be m_s the number of experimental data belonging to the I_s sub-interval and be $\hat{p}_s = m_s / N$ the probability estimate: $\sum_{s=1}^{S} m_s = N$, $\sum_{s=1}^{S} \hat{p}_s = 1$. The true (unknown) $p_s = \int_{I_s} p(x)dx$ probability that the investigated random variable takes values in the I_s interval belongs to the $(p_{1,s}; p_{2,s})$ interval with a confidence of $(1-\alpha)$:

$$p_{1,s} \cong \hat{p}_s - z_{\alpha/2}\sqrt{\hat{p}_s(1-\hat{p}_s)/N} \qquad p_{2,s} \cong \hat{p}_s + z_{\alpha/2}\sqrt{\hat{p}_s(1-\hat{p}_s)/N}, \quad (3)$$

where $z_{\alpha/2}$ is the $\alpha/2$ - point value of the Gaussian law of 0 mean and 1 variance. Note that (3) holds if and only if $N \cdot \hat{p}_s \cdot (1-\hat{p}_s) >> 1$, $s = 1,2,...S$.

2.3 Model Specification

This sub-section presents our original procedure designed in order to define a model for each of the $r = 1,2,...R$ information sources characterising a video watermarking attack. That is, by this procedure, a probability density function characterising the values taken by the noise vector can be obtained.

When trying to deploy the statistical tools presented above, the first problem is connected to the data dependency. Note that the components in the *noise* vector are computed on successive frames in the video sequence. Despite the mathematical properties of the DWT, the dependency among the frames in the video sequence can

be echoed in the *noise* vector components. Consequently, the statistical *iid* model required in *pdf* estimation is violated and we are to extract *iid* data by sampling the *noise* vector, *e.g.* with a large fixed period D. Actually, by shifting the sampling origin, D *iid* data sets, each of them with L/D components, can be obtained from the same *noise* vector, see Fig. 4.

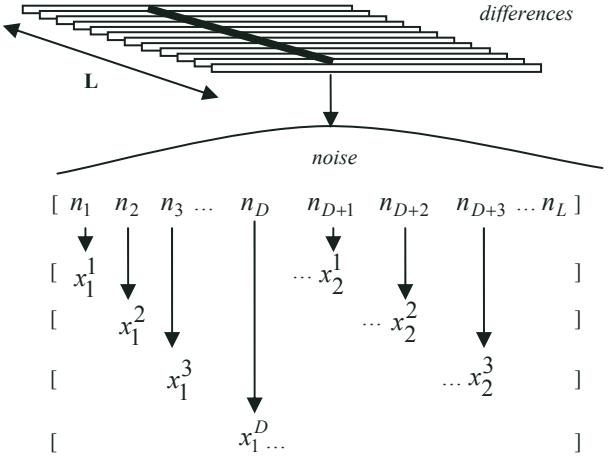


Fig. 4. By shifting the sampling period, several *iid* data sets can be obtained form the same *noise* vector

We may apply the EM Gaussian mixtures algorithm to each of these data sets and we may obtain D different *pdf* estimates. Which one is the best? Under the stationarity hypothesis, all these estimates provide the same information about the investigated attack. As we have no *a priori* information concerning stationarity, we should first verify the homogeneity of these estimates and then derive a selection criterion allowing us to state an unitary model. In this respect, we advance the following algorithm.

Algorithm 2

In order to model the attack effects on a rank (arbitrarily chosen) in the DWT coefficient hierarchy, the following 6 steps should be gone through:

Step 1: Periodically sample the *noise* vector, with a large enough period D. Shift the sampling origin and obtain the D *iid* data sets $[x^i{}_1, x^i{}_2, ..., x^i{}_N]$ ($i = 1,2,...,D$ and $N = L/D$), *cf.* Fig. 4.

Step 2: Apply the EM Gaussian mixture algorithm to each of the D *iid* data sets and obtain the D *pdf* estimates $\hat{p}^i(x)$, $i = 1,2,...,D$.

Step 3: Apply the EM Gaussian mixture algorithm to the whole *noise* vector and obtain the estimate $\hat{p}_{av}(x)$. Note that although $\hat{p}_{av}(x)$ has no mathematical support (it is computed on the whole noise vector, hence on dependent data), it expresses intuitively an average value for the $\hat{p}^i(x)$, $i = 1,2,...,D$ estimates.

Step 4: Define a similarity measure, denoted by $m(.)$, between two arbitrarily chosen $\hat{u}$ and $\hat{v}$ *pdf* estimates. Taking into account the discussion in Section 2.2, we considered a normalised sum of squared errors between probabilities, see (4):

$$m(\hat{u}(x),\hat{v}(x)) = \frac{\sum_{s=1}^{S}\left(\int_{I_s}\hat{u}(x)dx - \int_{I_s}\hat{v}(x)dx\right)^2}{\sum_{s=1}^{S}\left(\int_{I_s}\hat{v}(x)dx\right)^2}, \quad (4)$$

where I_s, $s=1,2,...S$ is a subdivision of the $[x_{\min};x_{\max}]$ interval on which the $\hat{u}$ and $\hat{v}$ functions take non-zero values. This similarity measure is not a distance because it is not symmetric; however, $m(\hat{u}(x),\hat{v}(x)) \geq 0$ and $m(\hat{u}(x),\hat{u}(x)) = 0$.

Step 5: **The model** for the investigated attack and rank will be that $\hat{p}^i(x)$ estimate to which $\hat{p}_{av}(x)$ is closest in the $m(.)$ sense; be it denoted by $\hat{p}(x)$:

$$\hat{p}(x) = \arg\min_{\hat{p}^i(x)} m(\hat{p}_{av}(x),\hat{p}^i(x)).$$

Step 6: In order to check up the pertinence of the model defined in the previous step, average the measures between each of the D estimates and the model:

$$Error = \frac{1}{D}\sum_{i=1}^{D} m(\hat{p}^i(x),\hat{p}(x)). \quad (5)$$

The *pdf* defined in *Step 5* can be considered as a true model only if the *Error* computed by (5) is lower than a prescribed upper limit (*e.g.* lower than 0.05).

Comments concerning Algorithm 2

In order to completely describe the attack effects, the previous algorithm should be applied for each investigated rank.

The model *per-se* will be computed in the next section. However, its validity is subject to further validations presented in Section 4.

3 Model Computation

The corpus considered in our experiments is composed of 10 video sequences, belonging to different movies, each of them about 25 minutes long ($L = 35000$ frames in each video sequence). Their content is heterogeneous, combining indoor and outdoor scenes, instable and arbitrary lighting conditions, still and high motion scenes. They are DivX coded at a low quality, according to mobile network requirements: a 64 Kbit/s rate, and frame sizes of 192 x 80 pixels.

The model we present in this section (Table 1) is computed on a single video sequence, arbitrarily chosen from the corpus. The rest of 9 video sequences are involved in the model evaluation process (Section 4).

The modelling refers to $R = 360$ largest coefficients of the (9,7) biorthogonal wavelet transform, applied at an $N_r = 3$ resolution level.

The investigation is carried out on four types of attacks: Gaussian filtering, sharpening, rotations (with 2 degrees), and StirMark.

Table 1. Statistical model for the watermarking attacks in the (9,7) DWT hierarchy

Attacks	Rank	Model parameters		Error
Gaussian filtering	$r = 1$	$P(k)$	0.0158 0.1990 0.1187 0.0750 0.0268 0.0461 0.0781 0.3378 0.0599 0.0424	0.0357
		$\mu(k)$	-0.0213 -0.0613 -0.0667 -0.0818 -0.1525 -0.0815 -0.0784 -0.0459 -0.1147 -0.0817	
		$\sigma(k)$	0.0005 0.0150 0.0148 0.0195 0.0228 0.0194 0.0185 0.0124 0.0030 0.0195	
	$r = 150$	$P(k)$	0.0158 0.1990 0.1187 0.0750 0.0268 0.0461 0.0781 0.3378 0.0599 0.0424	0.0170
		$\mu(k)$	-0.02136 -0.0613 -0.0667 -0.0818 -0.1525 -0.08158 -0.0784 -0.0459 -0.1147 -0.0817	
		$\sigma(k)$	0.0005 0.0150 0.0148 0.0195 0.0228 0.0194 0.0185 0.0124 0.0030 0.0195	
	$r = 300$	$P(k)$	0.02320 0.0583 0.1529 0.0439 0.3848 0.1173 0.1190 0.0784 0.0578 0.0711	0.0093
		$\mu(k)$	0.0292 0.0023 -0.0021 -0.0004 -0.0007 -0.0025 0.0591 -0.0021 0.0006 -0.0023	
		$\sigma(k)$	0.0148 0.0091 0.0026 0.0086 0.0016 0.0084 0.0108 0.0026 0.0086 0.0084	
Sharpening	$r = 1$	$P(k)$	0.0223 0.0235 0.107 0.06.74 0.1942 0.0144 0.1098 0.0866 0.1003 0.2738	0.0356
		$\mu(k)$	0.1009 -0.0290 0.0784 0.0712 0.0399 0.1887 0.1478 0.1577 0.1419 0.0585	
		$\sigma(k)$	0.0003 0.0006 0.1039 0.1036 0.0561 0.0022 0.0923 0.0898 0.0148 0.0515	
	$r = 150$	$P(k)$	0.0471 0.1325 0.1171 0.0413 0.2617 0.0613 0.0622 0.1014 0.1433 0.0154	0.0121
		$\mu(k)$	0.0444 0.0312 0.0315 0.0448 0.0037 0.0343 0.0419 0.0092 -0.0335 0.0454	
		$\sigma(k)$	0.0510 0.0063 0.0252 0.0509 0.0079 0.0504 0.0510 0.0385 0.0338 0.0509	
	$r = 300$	$P(k)$	0.4095 0.0159 0.1514 0.1137 0.0185 0.0527 0.0142 0.0726 0.0155 0.1355	0.0096
		$\mu(k)$	0.0013 0.03592 -0.0159 0.0289 -0.0405 -0.0109 0.0352 -0.0787 0.0436 0.0206	
		$\sigma(k)$	0.0039 0.0715 0.0098 0.0212 0.0436 0.0013 0.07161 0.0413 0.0703 0.0091	

Table 1. (*continued*)

Attacks	Rank	Model parameters		Error
Rotation 2°	$r = 1$	$P(k)$	0.1045 0.1024 0.0778 0.1091 0.1149 0.0980 0.1084 0.1002 0.0623 0.1218	0.0413
		$\mu(k)$	-0.4462 -0.5163 -0.5166 -0.7979 -0.3879 -0.1756 -0.9302 -0.5479 -0.6714 -0.0393	
		$\sigma(k)$	0.1856 0.2015 0.2017 0.2785 0.1596 0.1123 0.2669 0.2063 0.2311 0.0824	
	$r = 150$	$P(k)$	0.1015 0.0373 0.0384 0.0560 0.1059 0.0442 0.0447 0.3534 0.0315 0.1866	0.0150
		$\mu(k)$	-0.1247 -0.0740 -0.0331 0.0068 -0.1055 -0.06579 -0.0662 -0.0415 -0.0694 -0.0025	
		$\sigma(k)$	0.0800 0.1734 0.1624 0.1501 0.0843 0.1715 0.1716 0.0353 0.1724 0.0505	
	$r = 300$	$P(k)$	0.0197 0.0359 0.0735 0.3869 0.0269 0.0092 0.0155 0.3270 0.0365 0.0685	0.0096
		$\mu(k)$	0.1059 -0.0540 -0.0511 -0.0131 0.0842 0.1090 0.1013 -0.0050 -0.0347 -0.0449	
		$\sigma(k)$	0.1221 0.1595 0.0779 0.0298 0.1265 0.0401 0.1231 0.0070 0.1588 0.0793	
StirMark	$r = 1$	$P(k)$	0.0452 0.0972 0.1198 0.0163 0.09727 0.1015 0.0916 0.1219 0.1852 0.1235	0.0187
		$\mu(k)$	-0.0391 -0.2768 -0.1548 -0.5861 -0.01196 -0.1014 -0.0189 -0.0469 0.0053 -0.0647	
		$\sigma(k)$	0.1215 0.1457 0.0843 0.2525 0.0133 0.1403 0.1227 0.0617 0.0471 0.0604	
	$r = 150$	$P(k)$	0.2148 0.0812 0.0222 0.0430 0.2868 0.0457 0.1019 0.1382 0.0381 0.0276	0.0084
		$\mu(k)$	0.0014 0.0025 -0.0920 -0.0322 -0.0168 -0.0018 -0.0025 0.0032 -0.0201 -0.0547	
		$\sigma(k)$	0.0241 0.0677 0.0919 0.0533 0.0195 0.0672 0.0671 0.0022 0.0604 0.0898	
	$r = 300$	$P(k)$	0.0142 0.3721 0.0527 0.0233 0.0307 0.0071 0.2904 0.1810 0.0007 0.0272	0.0075
		$\mu(k)$	0.2002 0.0055 0.0564 -0.0597 -0.0397 -0.2174 -0.002 -0.0109 -0.0438 -0.0515	
		$\sigma(k)$	0.0264 0.0106 0.0171 0.0288 0.0358 0.0001 0.0003 0.0141 0.0351 0.0321	

When applying the *Algorithm 2*, the following numerical values are considered. The sampling period is $D = 250$ frames (*i.e.* 10s). $K = 10$ Gaussian laws are considered in the mixture and $N_{iter} = 200$ iterations are performed in the EM algorithm. In order to compute the model according to (4) and to evaluate the *Error* according to (5), $S = 20$ evenly distributed intervals are considered.

The model computation is illustrated in Table 1: we selected just three ranks ($r = 1, r = 150, r = 300$) but numerical values concerning the rest of ranks can be obtained by contacting the authors. Table 1 is organised according to the type of

attack and to the investigated rank; it provides the numerical values for the $P(k)$, μ_k, and σ_k parameters ($k = 1,2,...,K = 10$) which are to be considered in (1) in order to obtain the $\hat{p}(x)$ model. Fig. 5 draws these *pdf*s for the largest coefficients in two cases, namely Gaussian filtering and StirMark.

Table 1 also presents the average relative error in model computation (5): it can be seen that these values are very small (in each and every case lower than 0.05).

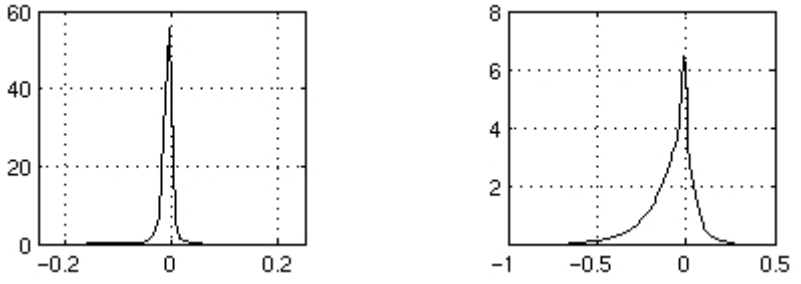


Fig. 5. The models computed for r=1 and for the Gaussian filtering (left) and StirMark (right)

4 Model Validation

The viability of any model is expressed by its ability to work in frameworks completely different from those in which it was computed. Consequently, we shall reconsider the key entities involved in the *Algorithm 2* and we shall verify whether the model provided in Table 1 still preserves its usefulness or not.

- *Is the model accuracy dependent on the similarity measure and error definition?*

The way in which the model was defined is intimately based on the similarity measure and error definitions expressed by (4) and (5). However, a good model should be able to describe all the D *pdf* estimates even when the similarity measure changes. As (4) expresses the differences between estimates in probability (*i.e.* between the integral on the *pdf*s), we are now to evaluate the differences between the *pdf*s themselves. In this respect, the $[x_{\min}; x_{\max}]$ domain is sampled in with a 0.001 step thus obtaining a set of x_t points. The new similarity measure between any of the $\hat{p}^i(x)$ estimates ($i = 1,2,...,D$) and the $\hat{p}(x)$ model is defined as a normalised sum of squared errors in the x_t points:

$$m'(\hat{p}^i(x), \hat{p}(x)) = \frac{\sum_t \left(\hat{p}^i(x_t) - \hat{p}(x_t)\right)^2}{\sum_t \left(\hat{p}(x_t)\right)^2}. \tag{4'}$$

Consequently, the new average error is computed according to (5'):

$$Error' = \frac{1}{D}\sum_{i=1}^{D} m'(\hat{p}^{\,i}(x), \hat{p}(x)) \,. \tag{5'}$$

The values of these new errors are displayed in Table 2. *As all these errors are quite small (lower than 0.1), with singular exceptions, we may consider them as a proof in favour of our model.*

Table 2. The average errors in model computation, when considering (5')

Atack	Rank $r = 1$	Rank $r = 150$	Rank $r = 300$
Gaussian filtering	0.0857	0.0395	0.0564
Sharpening	0.1098	0.0938	0.0772
Rotations 2°	0.0837	0.0547	0.0654
StirMark	0.0971	0.1707	0.0501

- *Is the model accuracy dependent on the EM Gaussian mixtures estimation?*

The final model (*i.e.* the $\hat{p}(x)$ function) should not depend on the estimation procedure. To check-out whether it is the case or not, we consider now probability estimation with confidence limits. We apply the procedure described in Section 2.2 for $S = 10$ evenly distributed subintervals of the $[x_{\min}; x_{\max}]$ interval. In order to obtain a value which does not depend on the particular *i.i.d.* data set, we compute the $\hat{p}_S$ estimates on the whole noise vector (*i.e.* on dependent data) and we apply (3) with $N = L/D$ and $\alpha = 0.05$ ($z_{\alpha/2} = 1.96$) .

We also compute the integral of the $\hat{p}(x)$ model on the same $S = 10$ evenly distributed division of the $[x_{\min}; x_{\max}]$ interval. In practically all cases (with very few exceptions), these probabilities computed starting form the model belong to the corresponding 95% confidence limits, as illustrated in Fig. 6. *We may consider that this is a very strong support for our model: the maximum likelihood criterion behind the EM estimation is somehow completed by a statistical confidence in the result.*

Note that when verifying the concordance between Gaussian mixture and confidence limit estimation, the test on the hypothesis that probability belongs to an interval

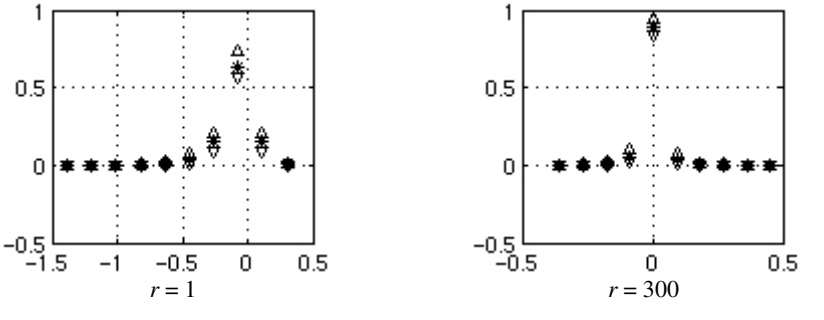


Fig. 6. Model validation (two ranks and the StirMark attack): the 95% confidence limits are represented by ∇ and Δ , while the probabilities computed from model are represented by *

[14] can be considered, thus associating a direct statistical confidence in the results. However, the criterion we took into consideration (verifying that the probability computed on the model belongs to the confidence interval) is more restrictive.

- *Is the model general or does it depend on the original video sequence?*

We are now to verify whether the model we provide in Table 1 is independent with respect to the video sequence on which it was computed.

In this respect, we consider the rest of 9 sequences from our corpus and we obtain their models, according to the *Algorithm 2*. In each of 9 cases, their inner errors, computed with (5) and (5'), are again very small (lower than 5%). Consequently, we can state that models for each investigated video sequence are successfully obtained.

Further on, we should verify whether these models are identical or not. In this respect, we evaluate the errors between each of these 9 new models and the reference model (the one in Table 1).

First, we computed the 9 errors between the 9 new models and the reference model, according to (4) and (5). The minimal, maximal, and mean values of these errors are denoted by Min_error, Max_error, and Av_error, respectively, see Table 3. We also computed the mean error when considering the (4') and (5') equations, see the column Av_error' (Table 3).

Table 3 refers just to the $r = 1$ rank, but we obtained the same type of results for all investigated ranks. *We can state now the generality (i.e. the independency with respect to the video sequence) of the model computed in Section 3.*

Table 3. The errors (average, minimal and maximal) between the reference models and the 9 models obtained on different video sequences. All the results correspond to $r = 1$.

Atack	Min_error	Max_error	Av_error	Av_error'
Gaussian filtering	0.0692	0.4972	0.2233	0.2978
Sharpening	0.0206	0.2121	0.1006	0.1005
Rotation 2°	0.0091	0.1474	0.0789	0.0763
StirMark	0.0048	0.0669	0.0278	0.0468

5 Conclusion and Future Work

This paper brings into evidence for the first time that the video watermarking attacks can be modelled by zero-memory information sources (one information source for each investigated rank) and provides the corresponding models (*pdf* approximating to these sources).

Note that the very existence of these information sources (including the stationarity validation) is an important result: it is supported by the fact that the same statistical model we computed does not depend either on the time origin or on the investigated video sequence. Moreover, maximum likelihood and confidence limit estimation pointed to the same model.

Beside all these assertions made by statistical tools, we also validate the model in a real-life watermarking application, in cooperation with our industrial partners. Note that for the same transparency and robustness, the speed of the insertion algorithm in [15] was increased up to 1000 times by considering a StirMark simulator according to the model in Table 1.

In perspective, we shall take advantage on the *Algorithm 2* generality and we shall apply it for other types of DWTs and attacks (FMLR, MPEG compressions, median filtering, …). We shall also adapt it for *pdf* estimation of the natural video sequences.

Such accurate models for attacks and/or original video are to allow significant advances in several challenging watermarking issues, as capacity evaluation or new detection rule designing, for instance.

Acknowledgments. This work is partially supported by the HD3D-IIO project of the Cap Digital competitiveness cluster.

References

1. http://www.iipa.com/special301_TOCs
2. Bangeman, E.: EU Commissioner criticizes iPod-iTunes tie-in, Ars Technica (March 11, 2007), http://arstechnica.com/news.ars/post/20070311-eu-commissioner-criticizes-ipod-itunes-tie-in.html
3. Fisher, K.: Musicload: 75% of customer service problems caused by DRM, Ars Technica (March 18, 2007), http://arstechnica.com/news.ars/post/20070318-75-percent-customer-problems-caused-by-drm.html
4. Cox, I., Miller, M., Bloom, J.: Digital Watermarking. Morgan Kaufmann Publishers, San Francisco (2002)
5. Arnold, M., Schmucker, M., Wolthusen, S.: Techniques and Applications of Digital Watermarking and Content Protection. Artech House (2003)
6. Mitrea, M., Duta, S., Zaharia, T., Prêteux, F.: Ensuring Multimedia Content Adaptability by Means of Data Hiding Techniques. In: Proc. SPIE, vol. 6383, 63830P61-63830P6 (2006)
7. Mitrea, M., Prêteux, F., Vlad, A., Fetita, C.: The 2D-DCT Coefficient Statistical Behaviour: A Comparative Analysis on Different Types of Image Sequences. Journal of Optoelectronics and Advanced Materials 6, 95–102 (2004)
8. Mitrea, M., Prêteux, F., Petrescu, M.: Very Low Bitrate Video: A Statistical Analysis in the DCT Domain. In: Atzori, L., Giusto, D.D., Leonardi, R., Pereira, F. (eds.) VLBV 2005. LNCS, vol. 3893, pp. 99–106. Springer, Heidelberg (2006)
9. Mitrea, M., Prêteux, F., Petrescu, M., Vlad, A.: The StirMark Watermarking Attack in the DWT Domain. In: Proc. 12th IEEE International Conference on Systems, Signals and Image Processing, Halkida, Greece, vol. 2, pp. 5–9 (2005)
10. Mitrea, M., Zaharia, T., Prêteux, F., Vlad, A.: Video Watermarking Based on Spread Spectrum and Wavelet Decomposition. In: Proc. SPIE, vol. 5607, pp. 156-164 (2005)
11. Mitrea, M., Dumitru, O., Prêteux, F.: Video Watermarking Capacity in the DWT Hierarchy. In: Proc. SPIE, Vol. 6576, 65760E1-65760E10 (2007)
12. Archambeau, C., Valle, M., Assenza, A., Verleysen, M.: Assessment of Probability Density Estimation Methods: Parzen Window and Finite Gaussian Mixtures. In: Proc. IEEE International Symposium on Circuits and Systems, Kos, Greece (May 2006)

13. Archambeau, C., Lee, J., Verleysen, M.: Convergence Problems of the EM Algorithm for Finite Gaussian Mixtures. In: Proc. 11th European Symposium on Artificial Neural Networks, Bruges, Belgium, pp. 99–106 (April 2003)
14. Vlad, A., Mitrea, A., Mitrea, M.: A Corpus – Based Analysis of How Accurately Printed Romanian Obeys Some Universal Laws. Wilson, A., Rayson, P., McEnery, T. (eds): A Rainbow of Corpora: Corpus Linguistics and the Languages of the World, Ch. 15, Lincom Europa Publishing House, pp. 153–165 (2003)
15. Mitrea, M., Prêteux, F., Nunez, J.: Procédé de Tatouage d'une Séquence Video, French Patent Request 05 54132, deposed on December 2005, in the name of SFR and GET/INT (2005)

On Statistical Independence in the Logistic Map: A Guide to Design New Chaotic Sequences Useful in Cryptography

Adriana Vlad[1,2], Adrian Luca[1], and Bogdan Badea[1]

[1] Faculty of Electronics, Telecommunications and Information Technology,
Politehnica University of Bucharest, 1-3, Iuliu Maniu Bvd.,
Bucharest 6, Romania
[2] The Research Institute for Artificial Intelligence,
Romanian Academy, 13, Calea 13 Septembrie, Bucharest 5, Romania
adriana_vlad@yahoo.com, avlad@racai.ro

Abstract. The paper explores the possibility to generate experimental *independent* and *identically distributed* data sets starting from the logistic map and discusses its applicability in cryptography. In order to reveal the statistical independence in the context of a chaotic signal, and to come up with firm and accurate results, the paper combines usual statistical methods with an original test procedure useful in the case of continuous random variables of unknown probability law. The overall theoretical approach may be viewed as a guide to generate *independent* and *identically distributed* samples starting from the logistic map, and also to design chaotic sequences useful for cryptographic purposes.

Keywords*:* chaotic behaviour, statistical independence, correlation coefficient, reference scatter diagram, chaotic enciphering sequences, natural text enciphering.

1 Introduction

In many engineering applications (see for example [1], Chapter 4) it is of great interest to dispose of chaotic sequences which comply with the *i.i.d.* statistical model (data coming out from *independent* and *identically distributed* random variables). Generating *i.i.d.* data sets brings into discussion both the stationarity issue, which enables identically distributed data, and the statistical independence problem. While the stationarity of the chaotic signals is generally assumed, [2], [3], the statistical independence seems to be in contradiction with the deterministic feature of chaotic signals. This aspect regarding the statistical independence of two random variables sampled from a chaotic signal is an important step required by the study. Based on the decision concerning the statistical independence we further come with new chaotic sequences which could be efficiently used in cryptography.

The statistical independence investigation and also the proposed enciphering methods are illustrated for the logistic map, (1), where R parameter belongs to the $(0;4)$ interval and x_n values belong to the $(0;1)$ interval. For a fixed R value, but different

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 460 – 474, 2007.
© Springer-Verlag Berlin Heidelberg 2007

x_0 initial conditions, an ensemble of trajectories is obtained. Each of these trajectories is considered a sample of the random process assigned to (1).

$$x_{n+1} = R\,x_n(1-x_n), \quad n = 0,1,2,... \tag{1}$$

Be two random variables sampled at two distinct iterations, n_1 and $n_2 = n_1 + d$, from the above mentioned random process. Our investigation has first to provide an answer whether the two random variables are statistically independent or not, and which will be the minimum d sampling distance which enables the statistical independence. The discussion on statistical independence will be carried out both in discrete case (that requires a non-linear transform applied to the random process assigned to the logistic map) and in continuous case.

Note that to check up the statistical independence of a chaotic signal is not a simple task. Although in the discrete case the statistical independence is more easily assumed, in the continuous case the answer is more difficult to foresee. For obtaining an accurate and firm decision concerning the possibility to generate discrete *i.i.d.* data sets, both investigations - in continuous and discrete case - are required.

Section 2 brings into discussion the continuous case, considering the two continuous random variables sampled from (1) at n_1 and $n_2 = n_1 + d$ iterations, without applying any transform. The decision regarding the statistical independence is based upon a new statistical procedure, [4], applicable to all kinds of continuous variables, normal or not, even of unknown statistical law. The new statistical test procedure includes and extends Spearman's Rank Correlation Test and permits the visual interpretation and decision concerning the independence of the two continuous random variables. The new test is required here because the experimental data are not normally distributed (in literature, there are well-known statistical tests applicable only to Gaussian populations). The analysis takes into account the effect of R parameter upon the statistical independence decision.

Section 3 presents the discrete case. This means that the random variables sampled at n_1 and $n_2 = n_1 + d$ iterations from the logistic map are first transformed into two discrete random variables, by applying a non-linear time invariant transform.

We mention that in the discrete case our investigation starts from the basic definition of statistical independence of two random variables (see for example [5]). The investigation of the statistical independence combines information theory tools with the Chi-square test used in contingency tables. A Monte Carlo analysis is also used to decide on the minimum d distance required to obtain statistical independence.

The results include the effect on the statistical independence of both the chosen non-linear transform and of the R parameter value of the logistic map.

Section 4 provides some modalities to construct new enciphering chaotic sequences using the *i.i.d.* discrete data sets generated in Sections 2 and 3. Examples are given to illustrate how suitable these chaotic sequences are for printed natural language enciphering.

2 A Study on Statistical Independence in the Continuous Case

We present our experimental study and decision regarding the statistical independence of the two continuous random variables, *X* and *Y*, sampled at n_1 and $n_2 = n_1 + d$ iterations, from the random process assigned to the logistic map. The analysis has to establish which will be the minimum *d* sampling distance leading to the acceptance of statistical independence between *X* and *Y*, and how this sampling distance is affected by the *R* parameter of the logistic map. To get an answer in the context of a chaotic signal, we used the test method related to Spearman's Rank Correlation Test, [4], applicable to all kind of continuous random variables, normal or not, even of unknown statistical law.

To facilitate the understanding of the experimental results, we briefly resume the test algorithm in [4]. The starting point is the acquisition of the two data samples of *N* size, corresponding to the two random variables, *X* and *Y*, each of the two data sets complying with the *i.i.d.* statistical model. The two data sets correspond to the two iterations n_1 and $n_2 = n_1 + d$ of the random process assigned to the logistic map:

$$x_1, x_2, \dots, x_i, \dots, x_N \qquad \text{and} \qquad y_1, y_2, \dots, y_i, \dots, y_N.$$

The notation (x_i, y_i) means a pair of values for *X* and *Y*, measured on the same trajectory at n_1 and $n_2 = n_1 + d$ iterations. Thus, there are *N* pairs in total, corresponding to the *N* trajectories generated by the logistic map.

The two hypotheses of the statistical test are: the null hypothesis, H_0, meaning that the random variables *X* and *Y* are statistically independent, and the alternative hypothesis, H_1, meaning that the random variables *X* and *Y* are dependent. The decision is based upon a statistical significance level $\alpha = 0.05$.

The independence test algorithm consists of the following main steps:

1. Sort x_i and y_i, data values in increasing order and obtain the experimental cumulative distribution functions (*cdf*) for *X* and for *Y*, $Fe_X(x)$ and $Fe_Y(y)$.

2. Apply $Fe_X(x)$ transform on the x_i data values and $Fe_Y(y)$ on the values y_i, $i = \overline{1, N}$, and obtain the new data x'_i and y'_i, which are uniformly distributed in $(0;1)$. By this data transform, Spearman's ranks are revealed.

3. Another random variable transform which is the inverse of the distribution function of the standard normal law, computed by numerical methods, is applied on data x'_i and y'_i, $i = \overline{1, N}$, generating the new data u_i and v_i. According to the known theory, both u_i and v_i values are standard normally distributed. Thus, two standard normal random variables *U* and *V* are obtained starting from the initial *X* and *Y*.

By this approach, we decide upon the independence of *X* and *Y* by testing the independence of *U* and *V*. This is a result of: *U* is a function only of *X* and *V* is a function only of *Y*. If *X* and *Y* are independent random variables, it results that *U* and *V* are also independent. As a first clear conclusion, if *U* and *V* are dependent, then the initial *X* and *Y* random variables are dependent.

Conversely, if *U* and *V* are independent we can think that *X* and *Y* are statistically independent. Note that the testing method applied in the continuous case requires a

very large amount of data for an accurate estimation of the cumulative distribution functions. Thus, in the context of the chaotic signal, the only decision to take in the continuous case is that X and Y "could be independent".

4. Compute r, the correlation coefficient between U and V, and then the t test value:

$$r = \sum_{i=1}^{N}(u_i - \overline{u})(v_i - \overline{v}) \Big/ \sqrt{\sum_{i=1}^{N}(u_i - \overline{u})^2 \sum_{i=1}^{N}(v_i - \overline{v})^2} \; ; \quad t = r\sqrt{(N-2) \Big/ (1-r^2)} \; . \tag{2}$$

If the null hypothesis H_0 is correct, then the test value should obey the Student law with N-2 degrees of freedom.

It can be easily shown that there is equality between r (from (2)) and b_1 (from (3)), where b_1 is the linear regression slope (V versus U):

$$b_1 = \sum_{i=1}^{N}\left(u_i v_i - \overline{u}\,\overline{v}\right) \Big/ \sum_{i=1}^{N}(u_i^2 - \overline{u}^2) \; . \tag{3}$$

5. Based on the α significance level, compute $t_{\alpha/2}$, the $\alpha/2$ point value of the Student law of N-2 degrees of freedom. If $|t| < t_{\alpha/2}$, then the two random variables X and Y could be independent. To decide upon independence, continue with step 6. Conversely, if $|t| \geq t_{\alpha/2}$, the random variables X and Y are dependent and no further investigation is required.

6. If $|t| < t_{\alpha/2}$, then test if U and V are jointly normally distributed. If U and V are proven jointly normally distributed, then we can think that X and Y are statistically independent. If U and V are not jointly normally distributed, then X and Y are dependent.

In order to decide upon the normal bivariate law, the visual inspection of the (u, v) *scatter diagram will be very useful: the (u, v) scatter diagram should be similar to that shown in Fig. 1 (the reference scatter diagram).*

The test procedure graphically brings into evidence that the bivariate normal law is essential when deciding upon statistical independence (each of the two random

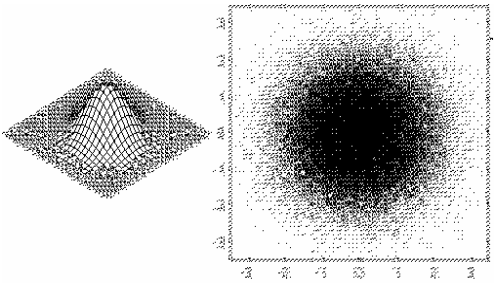


Fig. 1. The theoretical jointly normal probability density function and the corresponding scatter diagram for two independent random variables

variables U and V obeys the standard normal probability law, but not always the jointly normal probability law). The next experiments will point out this aspect. We next illustrate the statistical independence testing for two R values: 3.9 and 3.86. The scatter diagram and the linear regression slope are displayed on the same figure.

In Fig. 2 (a), for $R = 3.9$, $N = 40000$ data pairs and a distance $d = 10$ iterations, the test reveals dependent samples: $r = -0.0347 \Rightarrow t = -6.9483 \Rightarrow |t| > t_{\alpha/2}$ (computed by (2)); $\alpha = 0.05 \Rightarrow t_{\alpha/2} = 1.96$. In Fig. 2 (b), for $R = 3.9$, $N = 40000$ data pairs and $d = 30$, the test reveals independent samples: $r = -0.0001 \Rightarrow t = -0.0216 \Rightarrow |t| < t_{\alpha/2}$. If the scatter diagrams in Fig. 2 are compared to *the reference scatter diagram* from Fig. 1, it can be noticed that for $R = 3.9$ we can think of statistical independence only if $d = 30$ iterations.

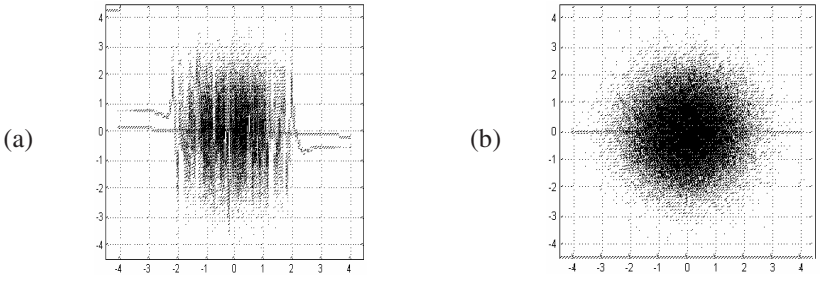


Fig. 2. Scatter diagrams (u, v) for $R = 3.9$ and a distance $d = 10$ (a) and $d = 30$ (b)

In Fig. 3, for $R = 3.86$ and a number of $N = 200000$ data pairs, the test decides dependent samples both for $d = 20$ where $r = -0.1109 \Rightarrow t = -49.8854 \Rightarrow |t| > t_{\alpha/2}$, and for $d = 30$, where $r = 0.1135 \Rightarrow t = 51.1098 \Rightarrow |t| > t_{\alpha/2}$.

If the scatter diagrams in Fig. 3 is compared to *the reference scatter diagram* in Fig. 1, it can be noticed that for $R = 3.86$ there is no statistical independence even for a sampling distance equal to 30 iterations. We obtained similar results for $R = 3.72$, *i.e.* there is no statistical independence for a sampling distance $d \leq 30$ iterations.

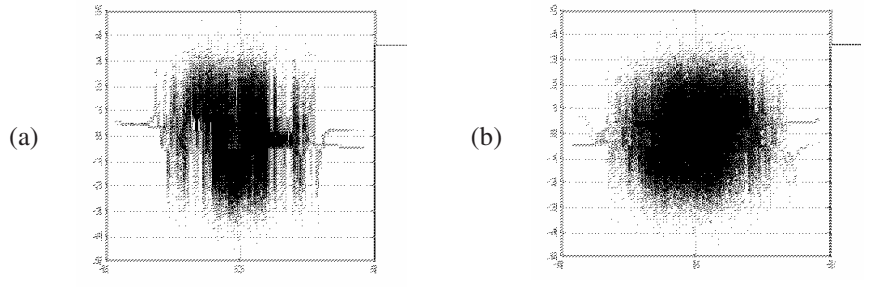


Fig. 3. Scatter diagrams (u, v) for $R = 3.86$ and a distance $d = 20$ (a) and $d = 30$ (b)

Conclusions for Section 2

In order to decide whether X and Y are dependent, the test procedure described in this section works well. Thus, for $R = 3.86$ and $R = 3.72$, the two continuous random variables are not statistically independent for $d \le 30$; the same conclusion is valid for $R = 3.9$ and $d < 20$ iterations. However, this does not imply to give up the investigation in the discrete case, Section 3. *The result in the discrete case cannot be foreseen only on the basis of the fact that the continuous investigation decided dependence*.

Certainly, the statistical independence in the continuous case implies statistical independence in the discrete case. However, if we are interested in obtaining discrete *i.i.d.* random variables we have to continue with the statistical investigation next described in Section 3, because the visual decision based upon scatter diagrams should be completed by means of numerical values. Moreover, the discrete case study could bring new information concerning the minimum d sampling distance for accepting statistical independence. This minimum sampling distance could be affected by the number of the distinct values of the discrete random variables, and also by the transform applied to obtain the discrete r.v. from the continuous one.

3 A Study on Statistical Independence in the Discrete Case

In this section, instead of the continuous random variable sampled from the random process assigned to (1), we shall consider a discrete random variable taking Q values. The Q discrete values are assigned to Q non-overlapping equal length intervals denoted by $I_i, i = \overline{1,Q}$ (see Figs. 4 and 5); these $I_i, i = \overline{1,Q}$ intervals cover the $(0;1)$ range of values taken by the logistic map. Each of the Q distinct values of the discrete random variable will have an occurrence probability equal to the probability that the logistic map takes values in the respective assigned I_i interval.

The investigation procedure is next illustrated for $Q = 6$. $Q = 6$ value was chosen having in view the possibility to encipher printed Romanian language with extended alphabet (including some orthography and punctuation marks, Section 4), and for a comparison with sequences produced by "the throwing of a die".

We next use two types of non-linear transforms from the continuous to the discrete case: either the *adjacent* transform or the *non-adjacent* transform, Fig. 4.

In the *adjacent* case, the Q discrete values are directly assigned to the Q *adjacent* non-overlapping equal length intervals, $I_i, i = \overline{1,Q}$. In the *non-adjacent* case, the Q discrete values of the random variable are again assigned to Q non-overlapping equal length intervals, I_i, $i = \overline{1,Q}$; this time, the I_i intervals cover the $(0;1)$ range of values in a different way. Thus, in the *non-adjacent* case, a subdivision of the $(0;1)$ interval in F non-overlapping equal length intervals is made first. Then, a new subdivision in Q non-overlapping equal length intervals is made for each of the F subintervals obtained before. Finally, each I_i interval is obtained as union of F subintervals as follows: $I_i = I_{i1} + I_{i2} + ... + I_{iF}$, $i = \overline{1,Q}$. Note that the *adjacent* case is in fact contained in the *non-adjacent* case for the division factor $F = 1$.

We next investigate the statistical independence of the two discrete random variables sampled from (1) at n_1 and $n_2 = n_1 + d$ iterations. In order to verify the statistical independence we shall simultaneously consider the two values taken by the logistic map at n_1 and n_2 iterations on the same trajectory. Finally, we will have N pairs of values obtained on the N different trajectories.

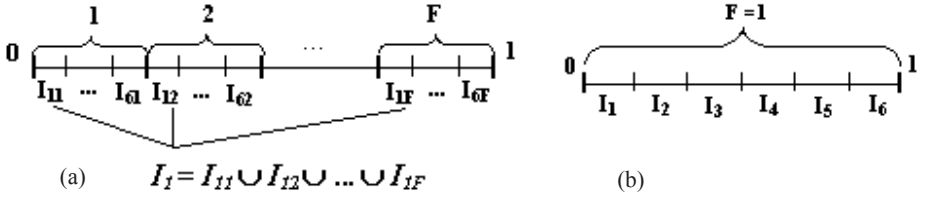


Fig. 4. (a) Obtaining the I_i intervals, $i = \overline{1,6}$, assigned to the discrete random variable in the *non-adjacent* case. F is the division factor of (0;1) interval. (b) For $F=1$ the *adjacent* case is met.

This leads to the noisy information channel (Fig. 6), where the input space $X = \{x_1,...,x_i,...,x_6\}$ corresponds to the I_i intervals, $i = \overline{1,6}$, at n_1 iteration, and the output space $Y = \{y_1,...,y_j,...,y_6\}$ corresponds to the I_j intervals, $j = \overline{1,6}$, at $n_2 = n_1 + d$ iteration. Hence, the two random variables under discussion for the statistical independence will be X and Y (with $Q = 6$ distinct values).

If the experimental mutual information, $I(X;Y)$, assigned to the information channel from Fig. 6 is zero, there is statistical independence between the output and the input of the noisy information channel, [6]. In fact, $I(X;Y) = 0$ means $p(x_i, y_j) = p(x_i)p(y_j)$, where $p(x_i, y_j)$ is the probability that a trajectory simultaneously passes through I_i and I_j intervals at n_1 and n_2 iterations, $p(x_i)$ is the probability that a trajectory passes through I_i interval at n_1 iteration, and $p(y_j)$ is the probability that a trajectory passes through I_j interval at n_2 iteration.

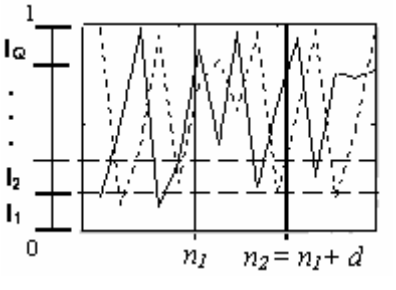


Fig. 5. Obtaining discrete random variables sampled at n_1 and n_2 iterations from the logistic map by using the *adjacent* non-linear transform

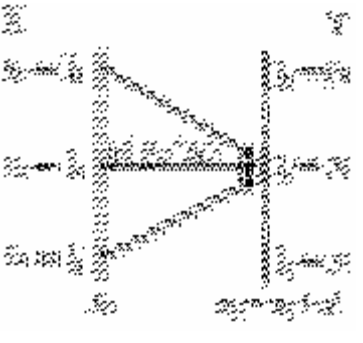


Fig. 6. The discrete information channel diagram

For the experimental results, the theoretical probabilities will be replaced by their estimated values: $\hat{p}(x_i, y_j) = m_{ij} / N$, where m_{ij} is the number of the investigated trajectories that simultaneously pass through I_i and I_j intervals; similarly, $\hat{p}(x_i) = m_i / N$ and $\hat{p}(y_j) = m_j / N$, where m_i (m_j) is the number of trajectories that pass through I_i (I_j) interval at n_1 (n_2) iteration.

The mutual information gives us rough numerical information concerning statistical independence, [7]. For the final decision we shall apply a Chi-square test used in contingency tables, [8], and a Monte Carlo analysis.

For the Chi-square test the two hypotheses are: the null hypothesis H_0, meaning to accept statistical independence of X and Y, *i.e.* $p(x_i, y_j) = p(x_i)p(y_j)$, $\forall i, j$; the alternative hypothesis H_1, meaning to reject the statistical independence of X and Y, *i.e.* $p(x_i, y_j) \neq p(x_i)p(y_j)$, $\forall i, j$. The test is based on the test value, (4):

$$z = \sum_{i=1}^{6}\sum_{j=1}^{6}\left(m_{ij} - \frac{m_i m_j}{N}\right)^2 \Bigg/ \left(\frac{m_i m_j}{N}\right). \tag{4}$$

If $z \leq z_\alpha$ (where z_α is the α point value corresponding to the Chi-square law of $n = (Q-1)^2 = 25$ degrees of freedom), we accept the null hypothesis H_0, *i.e.* we accept the statistical independence. In our illustrations, we considered the statistical significance level $\alpha = 0.05$, that leads to $z_\alpha = 37.652$.

We made a detailed numerical investigation concerning the effect of R parameter from (1), as well as of F division factor (from Fig. 4) on the statistical independence. The numerical results are obtained for different N - number of trajectories.

Table 1. The mutual information and the Chi-square test used in contingency tables

			N=50000		*N=150000*		*N=200000*	
			F=1	*F=100*	*F=1*	*F=100*	*F=1*	*F=100*
R=3.86	*d=15*	I(X;Y)	0.1435	0.00101	0.1486	0.00066	0.1467	0.00063
		H_0/H_1	H_1	H_1	H_1	H_1	H_1	H_1
	d=30	I(X;Y)	0.01905	0.00023	0.0203	0.00017	0.0192	0.00007
		H_0/H_1	H_1	H_0	H_1	H_0	H_1	H_0
R=3.9	*d=15*	I(X;Y)	0.00179	0.00037	0.00118	0.00031	0.00154	0.00025
		H_0/H_1	H_1	H_0	H_1	H_1	H_1	H_1
	d=30	I(X;Y)	0.00035	0.00046	0.00005	0.00011	0.00009	0.00006
		H_0/H_1	H_0	H_0	H_0	H_0	H_0	H_0

Some results are illustrated in Table 1. For example, for $R = 3.9$, $d = n_2 - n_1 = 15$ and $N = 50000$, in the *adjacent* case the mutual information was $I(X;Y) = 0.00179$. In the same case, for the Chi-square test used in contingency tables, the test value computed by (4) was $z = 52.2$ (larger than the point value $z_\alpha = 37.652$), so that the

test decision was H_1; hence statistical independence was rejected. For the same $R = 3.9$, $d = n_2 - n_1 = 15$ and $N = 50000$, but in the *non-adjacent* case ($F = 100$), the mutual information was $I(X;Y) = 0.00037$ and the test value was $z = 16.45$ (smaller than $z_\alpha = 37.652$), hence statistical independence was accepted.

From the above example we can think that the *F* division factor may affect the statistical independence. However, a final decision will be taken only after performing a Monte Carlo simulation for larger *N*- number of trajectories (see Tables 2).

From Table 1, it seems that *R* values affect statistical independence: see for example the results for $R = 3.86$ and $R = 3.9$ for $N = 200000$, $d = 30$ and $F = 1$.

Also, from Table 1 it seems that the statistical independence is no more affected by *R* values if the *non-adjacent* transform is used (see for example $F = 100$, $d = 30$ and $N = 200000$). However, when we continued the investigation by a Monte Carlo simulation and with larger *N* values, this conclusion was not valid any more.

The Chi-square test capabilities concerning the acceptance of wrong data as good data (*i.e.* to wrongly accept statistical independence) are evaluated by a Monte Carlo analysis. The test was resumed N_1 times for fixed *N* and different $d = n_2 - n_1$ values, and the proportion of the H_0 hypothesis acceptance was recorded. Tables 2 shows the results in the *adjacent* and *non-adjacent* cases for $N_1 = 500$, $R = 3.9$ and $d = 15$, $d = 20$ and $d = 30$. Table 2 also presents the same type of Monte Carlo analysis for $N_1 = 500$, $R = 3.86$ and $d = 30$, for *adjacent* and *non-adjacent* cases.

From Table 2, it can be noticed that for $R = 3.9$ and $F = 1$ there is a large proportion of H_0 hypothesis acceptance for a sampling distance $d = 30$ iterations; this proportion is stable as the number of investigated trajectories increases. For $d = 20$ iterations it is possible to wrongly accept statistical independence if the number of investigated trajectories is not large enough. When $R = 3.9$ and $F = 100$ we reached the same conclusion regarding the sampling distance as for $F = 1$: a sampling distance of $d = 30$ iterations enables to accept statistical independence. Note that for $F = 100$ and $d = 20$, a firm conclusion (meaning to reject statistical independence) was possible only for a large amount of data (this conclusion could not be foreseen even for $N = 100000$).

Table 2. Proportion of accepting H_0 hypothesis in the *adjacent* case (*F=1*) and in the *non-adjacent* case (*F=100*), for *R=3.9* and *R=3.86*

			N					
			5000	10000	50000	100000	200000	300000
R=3.9	*F*=1	*d*=15	0.682	0.33	0	0	0	0
		d=20	0.934	0.89	0.628	0.21	0	0
		d=30	0.931	0.946	0.952	0.955	0.945	0.946
	F=100	*d*=15	0.954	0.912	0.738	0.15	0.0460	0
		d=20	0.966	0.954	0.924	0.96	0.8900	0.77
		d=30	0.956	0.944	0.946	0.965	0.958	0.949
R=3.86	*F*=1	*d*=30	0.264	0.156	0	0	0	0
	F=100	*d*=30	0.952	0.934	0.901	0.884	0.672	0.564

From the same Table 2, it results that we cannot speak about statistical independence for $R = 3.86$ for a sampling distance $d = 30$, neither for $F = 1$ nor $F = 100$. *Note that these results in Table 2 could not be foreseen from Table 1.*

Conclusion for Section 3

A main quantitative result (supported by the two investigations, in the continuous and in the discrete case) is that we can speak about statistical independence in the context of a chaotic signal (here the logistic map). Thus, for $R = 3.9$ a sampling distance $d = 30$ is large enough to enable the statistical independence which, combined with the stationarity feature of the logistic map, justifies the *i.i.d.* data sequence generation. For the *non-adjacent* case (*e.g.* $F = 100$), these *i.i.d.* data sequences tend to observe the uniform distribution law (see Fig. 7 (b)).

From our experimental study, we can not say that the division factor F ($F = 1$ or $F = 100$) affects the final decision concerning the statistical independence. However, the probability distribution law corresponding to the random variables sampled from (1) is affected (see Fig. 7) and this aspect will be important in the construction of new chaotic sequences for cryptographic purposes (see Section 4).

On the other hand, both in the discrete and in the continuous case, the conclusion was that R parameter values affect the statistical independence decision, *e.g.* we can not speak about statistical independence for a sampling distance $d \le 30$ when $R = 3.86$ or $R = 3.72$.

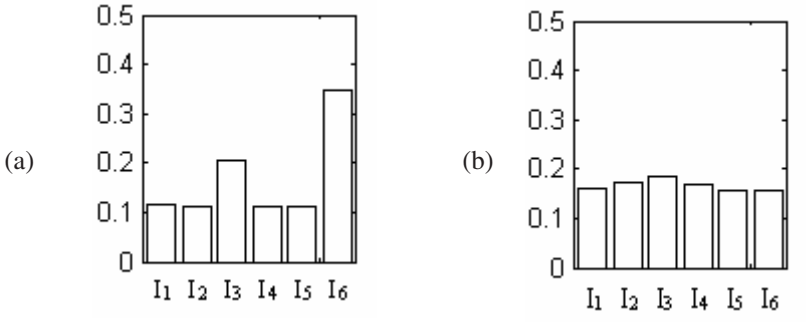


Fig. 7. Histograms for the relative frequencies distribution ($R = 3.9$ and $N = 40000$ trajectories): (a) *adjacent* (*F = 1*); (b) *non-adjacent* (for *F = 100*)

4 Constructing New Chaotic Sequences for Enciphering Natural Text

We discuss the possibility to construct new chaotic enciphering sequences based on the *i.i.d.* data sets previously obtained from the logistic map. We exemplify the usefulness of the *i.i.d.* discrete data sets in cryptography by enciphering printed Romanian language with $q = 36$ symbols in the alphabet (namely the most frequent 36 characters from the alphabet given in [9]). As we encipher natural text with $q = 36$ symbols, the initial chaotic sequence with $Q = 6$ distinct values is to be transformed

into a new discrete chaotic sequence with 36 distinct values; this is done by grouping the values from the initial chaotic sequence two by two.

Be $Z: z_0\ z_1\ \ldots z_n\ z_{n+1} \ldots z_{L-2}\ z_{L-1}$ a discrete chaotic sequence obtained from the logistic map as follows: each x_n continuous value is transformed into z_n discrete value according to Fig. 4 with $F=1$ or $F=100$. We consider $R=3.9$, a fixed randomly chosen x_0 initial condition in (1) and L successive iterations (without any sampling of the logistic map). Each z_n symbol from Z may take any of the $Q=6$ distinct values $0,1,\cdots,5$ assigned to the $I_1, I_2, \cdots, I_6$ intervals from Fig. 4.

Be $K: k_0\ k_1\ k_2\ \ldots$ an enciphering chaotic sequence with $q=36$ distinct symbols. Each k element in K is constructed by combining the symbols from Z two by two according to (5) and Fig. 8.

$$k = f(z_n, z_{n+r}) = 6\, z_{n+r} + z_n\,. \tag{5}$$

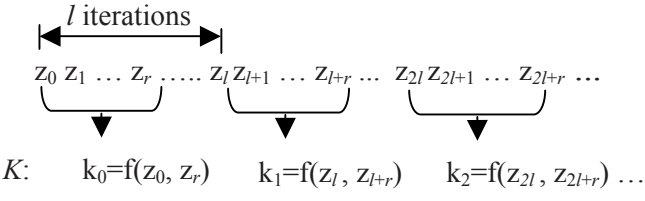


Fig. 8. The K enciphering chaotic sequence obtained by using (5)

The new random sequence K obeys different probability laws depending on r value (the number of iterations between the two involved z symbols, (5)). Note that l parameter does not affect the histograms of the 36 distinct values of K sequence (because the random process assigned to (1) is ergodic and the same is valid for Z and K). However, l parameter will play an important role in the m-gram structure of K sequence (m-gram means m successive characters in the respective sequence).

The next experimental results will also put into evidence the role of the division factor F in obtaining efficient enciphering chaotic sequences.

Fig. 9 plots the histograms of the 36 distinct values corresponding to four sequences: (a) the chaotic sequence K obtained as in Fig. 8, when $F=1$ and $r=30$; (b) the chaotic sequence K obtained as in Fig. 8, when $F=100$ and $r=30$; (c) a pseudorandom sequence obtained as in Fig. 8, when the Z sequence was generated by "the throwing of a loaded die" (the 6 probabilities assigned to the loaded die correspond to Fig. 7(a)); (d) a pseudorandom sequence obtained as in Fig. 8, when the Z sequence was generated by "the throwing of a fair die".

Our previous proof that a sampling distance of 30 iterations enables the obtaining of *i.i.d.* data sets from (1) is further valued and also supported by the investigation of the m-gram statistical structure. Thus, the K sequences which can be of interest for enciphering purposes correspond to $F=1$, $r \geq 30$ and $l>30$, or much better to $F=100$, $r \geq 5$ and $l>30$. By means of this method, we can generate chaotic sequences comparable with "the throwing of the die" type sequences.

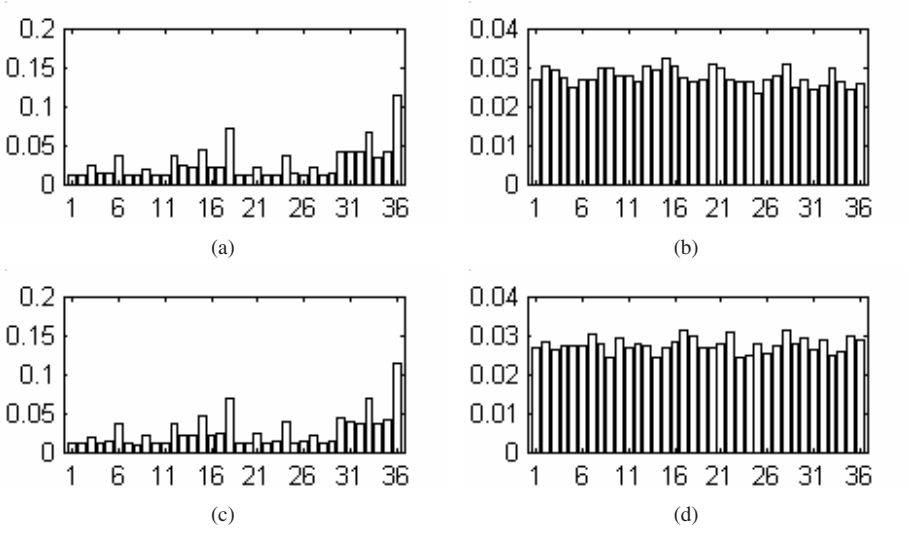


Fig. 9. Histograms for relative frequencies distribution of the q =36 symbols corresponding to: (a) *K* sequence for *F=1*, *r=30*; (b) *K* sequence for *F=100*, *r=30*; (c) *loaded die* sequence with the probabilities from Fig.7(a); (d) *fair die* sequence

The digram statistical structure assigned to the *K* sequences shows the effect of both *r* and *l* parameters in the quality of the enciphering chaotic sequences; this can be seen in Tables 3 and 4. The digrams are obtained from *K* sequence considering all the possible pairs of two successive symbols. Thus, for a *K* sequence of *N* symbols we will have *N*-1 digrams. In our illustrations, the size of *K* sequences was $N = 40000$ symbols. There are $q \times q = 36 \times 36 = 1296$ distinct possible digrams in total.

For the digram structure investigation we grouped the digrams into frequency classes according to their occurrence number and also depending on *F* division factor. Thus, for $F = 100$ we considered the following five frequency classes (Table 3): [0;20), [20;40), [40;60), [60;80) and [80;100]. For $F = 1$ we considered the following frequency classes (Table 4): [0], [1;20), [20;100), [100;200), [200;300) and [300;400].

How to read Table 3. In Table 3, for each frequency class, the number of distinct digrams (from the total 1296 possible distinct digrams) which belong to the respective class is indicated. For example, for $F = 100$, $r = 30$ and $l = 60$ there were 1180 distinct digrams having the occurrence number between 20 and 40; these 1180 digrams belong to the [20; 40] frequency class. When considering "the throwing of the fair die" there were 1177 distinct digrams in the same [20;40] frequency class.

Note that a very good result concerning the equalization of the diagram frequencies means to have many digrams in few frequency classes. Thus for $F = 100$, $r \geq 5$ and $l \geq 30$ (where we take advantage from the statistical independence investigation) we have obtained results which can be compared with "the throwing of a fair die", as we expected (Table 3).

Table 3. The *K* digram structure organized into frequency classes for *F=100*

	[0-20)	[20-40)	[40-60)	[60-80)	[80-100]
r =5, l =6	135	944	195	17	5
r =5, l =30	43	1123	129	1	0
r =30, l =60	19	1180	97	0	0
fair die	**24**	**1177**	**95**	**0**	**0**

Table 4 contains similar results as Table 3, but for the division factor $F = 1$. The first frequency class, [0], indicates the number of distinct digrams which did not occur in the sequence. Here again the statistical investigation study brings its contribution to the results (see the influence of *r* and *l* in the statistical digrams structure).

Table 4. The *K* digram structure organized into frequency classes for *F=1*

	[0]	[1-20)	[20-100)	[100-200)	[200-300)	[300-400]
r =5, l =6	905	23	250	73	23	22
r =5, l =30	1	687	550	39	14	5
r =30, l =60	0	661	565	61	4	5
loaded die	**0**	**672**	**552**	**62**	**6**	**4**

We next discuss some simple and practical enciphering methods using the above mentioned chaotic sequences. In our proposed enciphering method, the chaotic sequence (enciphering key) will be added modulo *q* (as in Vigenère type methods, [10]) to the plain text (original message) leading to the cryptogram (enciphered text).

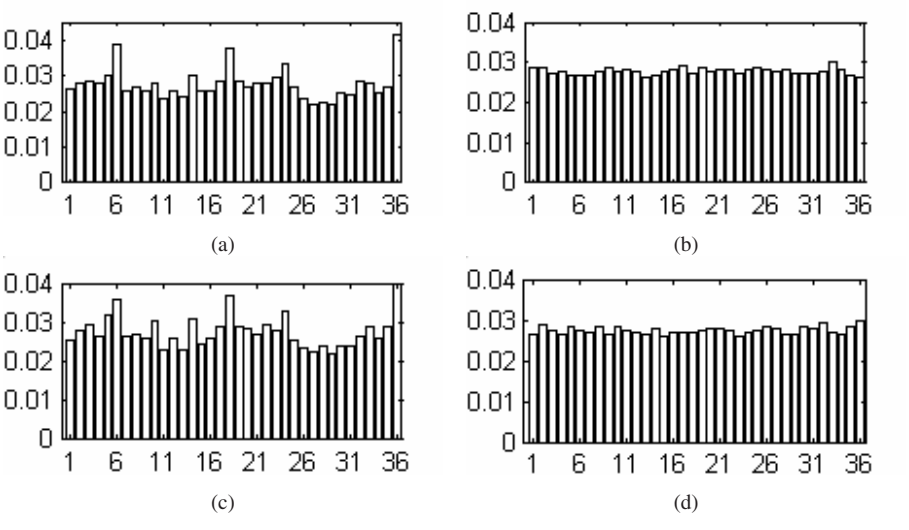


Fig. 10. Histograms for relative frequencies of the 36 distinct values of the cryptogram: (a)*F =1*, *r* = 30; (b) *F* =100, *r* =30; (c) *loaded die*; (d) *fair die*

To emphasize the usefulness of the chaotic sequences, we compare the following enciphering procedures: (a) adding to the natural text (printed Romanian) a chaotic sequence of *K* type, as long as the natural text; (b) adding to the natural text (printed Romanian) a sequence obtained by "the throwing of a die", sequence as long as the natural text. For the second type cipher, depending on F value, $F=1$ or $F=100$, we shall consider "the throwing of a loaded die" or "the throwing of a fair die", respectively. The latter means to simulate the Vernam cipher.

Fig. 10 plots the histograms of the relative frequencies distribution of the $q=36$ distinct values in the cryptogram, corresponding to the above-mentioned ciphers.

Figs. 10 (a) and (b) refer to the frequency profile cryptogram obtained by adding a chaotic sequence of *K* type to the natural text (printed Romanian). Figs. 10 (c) and (d) correspond to the frequency profile cryptogram obtained by adding a sequence generated by "the throwing of a die" to the natural text (printed Romanian).

The cryptogram digram structure corresponding to the above mentioned ciphers is presented in Table 5; this table is organized in a similar way as Table 3.

Table 5. The cryptogram digram structure organized into frequency classes for *F=1* and *F=100*

		[0-20)	[20-40)	[40-60)	[60-80)	[80-100]
F=1	$r=5, l=6$	166	884	223	22	1
	$r=5, l=30$	106	989	179	21	1
	$r=30, l=60$	83	1016	189	8	0
	loaded die	**71**	**1046**	**175**	**4**	**0**
F=100	$r=5, l=6$	15	1188	93	0	0
	$r=5, l=30$	23	1183	90	0	0
	$r=30, l=60$	14	1205	77	0	0
	fair die	**20**	**1195**	**81**	**0**	**0**

The overall results bring into evidence the possibility to design efficient enciphering methods using the chaotic *i.i.d.* sequences generated from (1).

5 Conclusions

The method of investigation and decision-making concerning statistical independence in the context of the logistic map allows the obtaining of new *i.i.d.* chaotic sequences of a large applicability in various fields. All the investigation was made with a statistical error control, providing firm and accurate quantitative results. We next used the obtained numerical results to construct new *i.i.d.* chaotic sequences useful in cryptography.

The paper suggests and proves the utility of these chaotic sequences as keys in various cryptosystems. An immediate example, useful in different variants, is to perform a summation modulo q successively for each character, between the plaintext and the key (q is the number of distinct characters in the alphabet of the plaintext). On the basis of the redundancy of the information source corresponding to the key, and also using some knowledge about the natural language model, the performance of the cipher can be evaluated.

Acknowledgement. This paper was in part supported by a Grant of The National University Research Council – CNCSIS, 89/2006-2007, Romania.

References

1. Kennedy, M.P., Rovatti, R., Setti, G.: Chaotic Electronics in Telecommunications. CRC Press, Boca Raton (2000)
2. Lasota, A., Mackey, M.C.: Chaos, Fractals, and Noise. Stochastic Aspects of Dynamics, 2nd edn. Springer, Heidelberg (1994)
3. Baptista, M.S.: Cryptography with chaos. Physics Letters A 240, 50–54 (1998)
4. Badea, B., Vlad, A.: Revealing statistical independence of two experimental data sets. An improvement on Spearman's algorithm. In: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganà, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3980, pp. 1166–1176. Springer, Heidelberg (2006)
5. Papoulis, A.: Probability, Random Variables, and Stochastic Processes. McGraw-Hill, New York (1965)
6. Shannon, C.E.: A Mathematical Theory of Communication. Bell Syst. Tech. J. 27, 379–423, 623-656 (1948)
7. Luca, A., Vlad, A.: Generating Identically and Independently Distributed Samples Starting from Chaotic Signals. In: Proc. IEEE Int. Symposium on Signals, Circuits and Systems (ISSCS 2005), Iaşi, Romania, pp. 227–230 (2005)
8. Walpole, R.E., Myers, R.H.: Probability and Statistics for Engineers Scientists, 4th edn. MacMillan Publishing, New York (1989)
9. Vlad, A., Mitrea, A., Mitrea, M.: Orthography and Punctuation Marks in Printed Romanian Modelling: a Study Based on the Letter Statistical Structure in a Literary Corpus. In: Burileanu, C., Teodorescu, H.-N.(eds.): Advances in Spoken Language Technology. The Publishing House of the Romanian Academy, Bucharest, pp. 207–216 (2007)
10. Shannon, C.E.: Communication Theory of Secrecy Systems. Bell Systems Technical Journal 28, 656–715 (1949)

FVM- and FEM-Solution of Elliptical Boundary Value Problems in Different Coordinate Systems

Günter Bärwolff

Technische Universität Berlin, Fak. II, Inst.f.Mathematik,
Str. des 17. Juni 136, D-10623 Berlin, Germany
baerwolf@math.tu-berlin.de
http://www.math.tu-berlin.de/~baerwolf

Abstract. In this paper several numerical concepts for the solution of elliptical boundary value problems will be compared with regard to their efficiency, computational load and accuracy. Especially regions which are images of rectangles and cuboids of coordinate transformations will be considered. I argue that it is thus possible to construct well structured discretisations which imply equation systems with very high solution properties. The approaches discussed will be tested with instructive examples in the light of their advantages and disadvantages.

Keywords: FEM, FVM, elliptical boundary value problem, weak formulation, flux balance, concave edged region, Cartesian and polar coordinates, ams-classification: 65N06, 65N30, 65N50, 65Y20.

1 Introduction

In physics, engineering fields and applied mathematics there are mathematical models deployed which include for example parabolic or elliptic differential equations. In fluid dynamics the mass and momentum balance lead to the Navier-Stokes equation and the Maxwell equations as a result of the modeling of the electrical and magnetic field.

The initial boundary value problems (non-stationary) or the boundary value problems (stationary) mostly must be solved numerically due to occurring geometries, boundary conditions or complicated source terms.

Elliptic differential equations of the form

$$-\Delta u + \alpha u = f \quad \text{in } \Omega \tag{1}$$

are often to solve as important subproblems of complicated and nested numerical solution algorithms for the whole initial boundary value problems.

Considering the stationary Navier-Stokes equation (for example, see [1])

$$(v \cdot \nabla)v = -\nabla p + \nu \Delta v + f \tag{2}$$

$$\nabla \cdot v = 0 \tag{3}$$

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 475–485, 2007.
© Springer-Verlag Berlin Heidelberg 2007

and applying (3) to (2), we get a Poisson equation for the pressure p

$$-\Delta p = \nabla \cdot (v \cdot \nabla)v - \nabla \cdot f \ .$$

Thus it is possible to formulate an iteration method

$$-\Delta p = \nabla \cdot (v^n \cdot \nabla)v^n - \nabla \cdot f^n \tag{4}$$

$$-\nu \Delta v^{n+1} = -\nabla p \tag{5}$$

to update the velocity field v^n to v^{n+1}. The iteration method (4), (5) means the solution of three (2d case) or four (3d case) equations of type (1).

As boundary conditions for the equation (1), we consider Dirichlet data and Robin type conditions. By appropriate homogenisation we can transform non-homogeneous Dirichlet data to an auxiliary problem with homogeneous boundary conditions of first order. For our further discussion, we generalise (1) to the boundary value value problem

$$-\nabla \cdot (\lambda \nabla u) + \alpha u = f \quad \text{in } \Omega \tag{6}$$

$$u = 0 \quad \text{on } \Gamma_d \ , \quad \lambda \partial_\nu u + \kappa u = g \quad \text{on } \Gamma_n \tag{7}$$

with real parameters or continuous functions $\lambda > 0$, $\alpha \geq 0$, $\kappa \geq 0$, g, f. For the boundary $\Gamma = \Gamma_d \cup \Gamma_n$ of $\Omega \subset \mathbb{R}^n$, $n = 2, 3$, we have $\Gamma_d \cap \Gamma_n = \emptyset$.

2 Functional Analytical Setting and Numerical Methods

2.1 Finite-Element Method (FEM)

For the functional analytical setting of the boundary value problem (6), (7) we consider the space

$$U = \{u \in H^1(\Omega),\ u = 0 \text{ on } \Gamma_d\}$$

as a solution space and $V = U$ as a test space. With piecewise linear hat functions (finite elements) $\phi_1, \ldots, \phi_N$, we define a finite dimensional subspace $V_h \subset V$

$$V_n = \text{span}\{\psi_1, \psi_2, \ldots, \psi_N\} \ .$$

Multiplying equation (6) with an element $v \in V$ and integrating over Ω we find

$$-\int_\Omega \nabla \cdot (\lambda \nabla u) v \, d\omega + \int_\Omega \alpha u v \, d\omega = \int_\Omega f v \, d\omega$$

and integration by parts leads to

$$\int_\Omega \lambda \nabla u \cdot \nabla v \, d\omega - \int_\Gamma \lambda \partial_\nu u \, v \, ds = \int_\Omega f v \, d\omega \ .$$

Using (7) and the property $v = 0|_{\Gamma_d}$ yields the weak formulation of problem (6), (7)

$$\int_\Omega \lambda \nabla u \cdot \nabla v \, d\omega + \int_{\Gamma_n} \kappa u \, v \, ds = \int_\Omega f v \, d\omega + \int_{\Gamma_n} g \, v \, ds \ . \tag{8}$$

With the definitions

$$A(u,v) := \int_\Omega \lambda \nabla u \cdot \nabla v \, d\omega + \int_{\Gamma_n} \alpha u \, v \, ds \, , \quad F(v) := \int_\Omega f v \, d\omega + \int_{\Gamma_n} g \, v \, ds + \int_\Omega \alpha u v \, d\omega,$$

we can establish (8) as

$$A(u,v) = F(v) \, , \quad v \in V \, , \tag{9}$$

with the symmetric bilinear form $A : V \times V \to \mathbb{R}$ and the linear functional $F : V \to \mathbb{R}$. Under the given assumptions about the right hand side f and the other parameter functions, there exists a unique solution $u \in U$ of (9) which we call the weak solution of problem (6), (7) (s. for example [4]). If we have $u \in C^2(\Omega)$ (twice continuous differentiable), the weak solution u is also a solution of the original boundary value problem (6), (7). On the other hand, a solution of the original problem is also a weak solution.

For the numerical solution we make the ansatz

$$u_h(\mathbf{x}) = \sum_{j=1}^{N} c_j \psi_j(\mathbf{x}) \tag{10}$$

and using (9), we find for $i = 1, \ldots, N$ with

$$\begin{aligned} A(u_h, \psi_i) = F(\psi_i) &\Longleftrightarrow A(\sum_{j=1}^{N} c_j \psi_j, \psi_i) = F(\psi_i) \\ &\Longleftrightarrow \sum_{j=1}^{N} A(\psi_j, \psi_i) c_j = F(\psi_i) \Longleftrightarrow \mathcal{A}\mathbf{c} = \mathcal{F} \end{aligned} \tag{11}$$

a linear equation system with the coefficient matrix $\mathcal{A} = (A(\psi_i, \psi_j)_{ij})$, the right hand side $\mathcal{F} = (F(\psi_1), \ldots, F(\psi_N))^T$ and $\mathbf{c} = (c_1, \ldots, c_N)^T$. Depending on the choice of the finite elements ϕ_i or the space V_h respectively, we get a sparse matrix $\mathcal{A}$. If we have a smoothness $u \in H^p(\Omega)$, $p \geq 2$ for the solution u and linear finite elements for convergence, the estimate

$$\|u_h - u\| \leq K \, h \|u\|_{H^2}$$

is valid (h beeing the characteristic finite-element grid parameter).

2.2 Finite-Volume Method (FVM)

For the finite-volume method we divide the region Ω into small subdomains ω_k, i.e. $\Omega = \bigcup_{k=1}^{N} \omega_i$. We now consider equation (6) in the control volume ω_k by integrating

$$\int_{\omega_k} [-\nabla \cdot (\lambda \nabla u) + \kappa u] \, d\omega = \int_{\omega_k} f \, d\omega \, . \tag{12}$$

Using the theorem of Gauß from (12), we have

$$-\int_{\gamma_k} \lambda \partial_\nu u \, ds + \int_{\omega_k} \alpha u \, d\omega = \int_{\omega_k} f \, d\omega \; , \tag{13}$$

with the boundary γ_k of ω_k. If, in the 2d case, we use rectangular control volumes with the length Δx and the heigth Δy we can approximate equation (13) by

$$\begin{aligned} -\lambda_{i+1/2j}\frac{u_{i+1j}-u_{ij}}{\Delta x}\Delta y + \lambda_{i-1/2j}\frac{u_{ij}-u_{i-1j}}{\Delta x}\Delta y \\ -\lambda_{ij+1/2}\frac{u_{ij+1}-u_{ij}}{\Delta y}\Delta x + \lambda_{ij-1/2}\frac{u_{ij}-u_{ij-1}}{\Delta y}\Delta x + \alpha_{ij}u_{ij}\Delta x\Delta y = f_{ij}\Delta x\Delta y \end{aligned}$$

where (ij) denotes the center position of ω_k, $(i+1/2j)$ the position on the right part of boundary γ_k, and so on. After dividing by $\Delta x\Delta y$ we get for the finite volume ω_k the discrete equation

$$\begin{aligned} [(\lambda_{i+1/2j}+\lambda_{i-1/2j})/\Delta x^2 + (\lambda_{ij+1/2}+\lambda_{ij-1/2})/\Delta y^2 + \kappa_{ij}]u_{ij} \\ -\lambda_{i+1/2j}u_{i+1j}/\Delta x^2 - \lambda_{i-1/2j}u_{i-1j}/\Delta x^2 \\ -\lambda_{ij+1/2}u_{ij+1}/\Delta y^2 - \lambda_{ij-1/2}u_{ij-1}/\Delta y^2 + \alpha_(ij)u_(ij) = f_{ij} \; . \end{aligned} \tag{14}$$

Equation (14) $(k = 1, \ldots, N)$ together with an appropriate approximation of the boundary conditions (7) gives an equation system for the determination of the finite-volume solution u_{ij}.

If we assume enough smoothness of the solution u of (6), (7) for example $u \in C^3(\Omega)$, we can show that (14) is an approximation of (6) of second order and the convergence result

$$\|u_{ij} - u(x_{ij}\|_\infty \leq K\, h^2$$

holds.

2.3 Coordinate Transformation of the Boundary Value Problem

If region Ω is a circle or a certain part of a circle or cylinder, it is possible in most cases to represent Ω as an image of a transformation. Let Ω for example be

$$\Omega = \{(x,y) \in \mathbb{R}^2,\, x^2+y^2 \leq R^2\} \setminus \{(x,y) \in \mathbb{R}^2, x \leq 0,\, x \leq y \leq -x\} \; ,$$

there is a transformation $\mathbf{x} : [0,R] \times [-3\pi/4, 3\pi/4] \to \Omega$

$$\mathbf{x}(r,\phi) = \begin{pmatrix} x(r,\phi) \\ y(r,\phi) \end{pmatrix} = \begin{pmatrix} r\cos\phi \\ r\sin\phi \end{pmatrix} \quad \text{with} \quad \mathbf{x}([0,R]\times[-3\pi/4,3\pi/4]) = \Omega \; .$$

Introducing

$$\hat{u}(r,\phi) = u(x(r,\phi), y(r,\phi)) \; ,$$

we can transform the weak formulation (8) to

$$\int_{\hat{\Omega}} \hat{\lambda}\hat{\nabla}\hat{u}\cdot\hat{\nabla}\hat{v}\, r d\phi dr + \int_{\hat{\Gamma}_n} \hat{\kappa}\hat{u}\,\hat{v}\, r\, d\phi dr = \int_{\hat{\Omega}} \hat{\alpha}\hat{u}\hat{v}\, r\, d\phi dr + \int_{\hat{\Gamma}_n} \hat{g}\,\hat{v}\, r\, d\phi dr$$

or

$$\int_{\hat{\Omega}} \hat{\lambda}(r\frac{\partial\hat{u}}{\partial r}\frac{\partial\hat{v}}{\partial r} + \frac{1}{r}\frac{\partial\hat{u}}{\partial\phi}\frac{\partial\hat{v}}{\partial\phi})r\, d\phi dr + \int_{\hat{\Gamma}_n} \hat{\kappa}\hat{u}\,\hat{v}\, r\, d\phi dr = \int_{\hat{\Omega}} \hat{\alpha}\hat{u}\hat{v}\, r\, d\phi dr + \int_{\hat{\Gamma}_n} \hat{g}\,\hat{v}\, r\, d\phi dr \tag{15}$$

with the rectangle $\hat{\Omega} = [0, R]\times[-3\pi/4, 3\pi/4]$. The advantage of the transformed weak formulation lies in the very simple integration region $\hat{\Omega}$. Thus a discretisation of $\hat{\Omega}$ can be attained in a very regular and structured manner.

We are now going to transform the equations (12), (13) as a base for the finite-volume method. We have

$$\int_{\hat{\omega}_k} [-\hat{\nabla}\cdot(\hat{\lambda}\hat{\nabla}\hat{u}) + \hat{\alpha}\hat{u}]\, r\, d\phi dr = \int_{\hat{\omega}_k} \hat{f}\, r\, d\phi dr\ . \tag{16}$$

Using the theorem of Gauß from (16), there follows

$$-\int_{\hat{\gamma}_k} \hat{\lambda}\partial_\nu\hat{u}\, d\hat{s} + \int_{\hat{\omega}_k} \hat{\alpha}\hat{u}\, r\, d\phi dr = \int_{\hat{\omega}_k} \hat{f}\, r\, d\phi dr\ , \tag{17}$$

where $\hat{\omega}_k = [r_i, r_i + \Delta r]\times[\phi_j, \phi_j + \Delta\phi]$ is a subrectangle of $\hat{\Omega}$. $d\hat{s}$ means in the r-direction dr, and in the ϕ-direction $r\, d\phi$.

It is important to note that the transformed equation (1)

$$-[\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial\hat{u}}{\partial r}) + \frac{1}{r^2}\frac{\partial^2\hat{u}}{\partial\phi^2}] + \hat{\alpha}\hat{u} = \hat{f} \quad \text{on } \hat{\Omega}$$

cannot be used to create a finite-volume method by making a balance of the form

$$-\int_{\hat{\omega}_k} [\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial\hat{u}}{\partial r}) + \frac{1}{r^2}\frac{\partial^2\hat{u}}{\partial\phi^2} - \hat{\alpha}\hat{u}]d\phi dr = \int_{\hat{\omega}_k} \hat{f}\, d\phi dr\ , \tag{18}$$

because the area element $d\omega$ is equal to $r\, d\phi dr$ and not equal to $d\phi dr$. (18) is not a balance over a control area (or control volume in the 3d case) compared to (17). In the language of engineering, the integral

$$\int_{\hat{\gamma}_k} \hat{\lambda}\partial_\nu\hat{u}\, d\hat{s}$$

of the balance (18) is in the 2d case a flux through an edge of the control area $\hat{\omega}_k$ which implies that $d\hat{s}$ must be a length (in the 3d case we have a flux through the surface of a control volume).

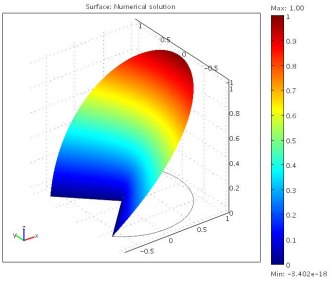


Fig. 1. Numerical solution of (19) - 3488 degrees of freedom

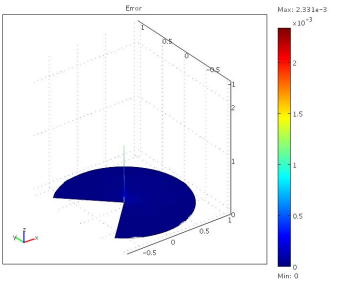


Fig. 2. Error - 3488 degrees of freedom

3 Numerical Examples

Let us consider the boundary value problem

$$-\Delta u = 0 \quad \text{in } \Omega, \quad u(x,y) = \cos(\frac{2}{3}\arctan(\frac{y}{x})) \quad \text{on } \Gamma = \partial\Omega \tag{19}$$

with

$$\Omega = \{(x,y) \in \mathbb{R}^2,\, x^2+y^2 \le 1\} \setminus \{(x,y) \in \mathbb{R}^2, x \le 0,\, x \le y \le -x\}$$

as a circle sector including a singular boundary point $\mathbf{x} = (0,0)$, i.e., a blunt angle and a concave region. We know with

$$u(x,y) = (x^2+y^2)^{1/3}\cos(\frac{2}{3}\arctan(\frac{y}{x}))$$

the exact analytic solution of problem (19). In polar coordinates we can write the solution in the form

$$\hat{u}(r,\phi) = r^{2/3}\cos(\frac{2}{3}\phi)$$

for $(r,\phi) \in [0,1] \times [-3\pi/4, 3\pi/4]$.

The use of the standard FEM-code COMSOL (see [2]) yields the numerical solution and the error shown in Figures 1, 2 on the adaptive grid shown in Figure 3 (drawn over Ω).

Figures 1 - 4 show the influence of the singular point $\mathbf{x} = (0,0)$ to the error distribution (see also Table 1). Also, a strong refinement leads to a small decrease of the error near the singular point. We observe an increase in computational load because of the vast number of elements and degrees of freedom respectively, yet without a significant reduction of the error.

Using the transformed weak formulation (15) with structured triangle or quadrilateral FEM grids we can reduce the error and the computational load significantly. Because of the continuous differentiability of the transformation $\mathbf{x}$ and the injectivity for inner points of $\hat{\Omega}$ piecewise linear hat functions $\hat{\psi}_k$ on $\hat{\Omega}$

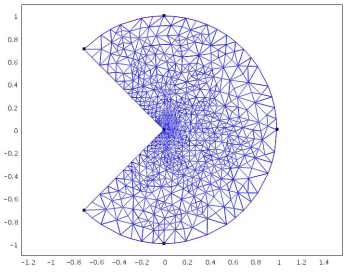


Fig. 3. Adapted triangular grid - 3488 degrees of freedom

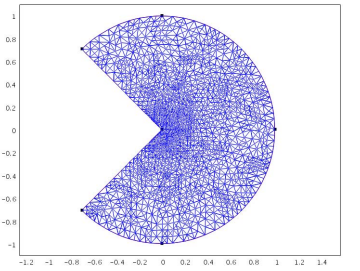


Fig. 4. Refined Grid - 8369 degrees of freedom

Table 1. FEM results of Cartesian coordinates

Method	Grid	Maximal error	degrees of freedom	Figure
FEM	adapted, unstructured	$2.331e-03$	3488	1-3
FEM	adapted, unstructured	$9.952e-04$	8369	4

have images $\psi(\mathbf{x}(r,\phi)) = \hat{\psi}_k(r,\phi)$ which are admissible. Thus the convergence results of the numerical solution of problem (8) in Cartesian coordinates will be transformed to the FEM solution of problem (15). Figures 5-8 show the results plotted over the region $\hat{\Omega} = \mathbf{x}^{-1}(\Omega)$, i.e., the singular point $\mathbf{x} = (0,0) \in \Gamma$ degenerates to the edge $\{(0,\phi),\ -3\pi/4 \leq \phi \leq 3\pi/4\}$ of the region $\hat{\Omega}$. The results will be summarised in Table 2.

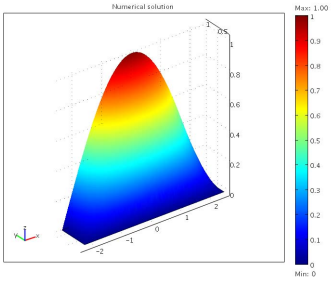


Fig. 5. Numerical solution of (19) using polar coordinates - 9062 degrees of freedom

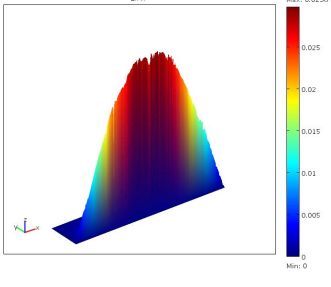


Fig. 6. Error - 9062 degrees of freedom

With regard to the finite-element results, we now consider the numerical solution by a finite-volume method. Starting with the balance equation (17), we construct a finite-volume method for a rectangular grid of $\hat{\Omega}$ following [3], [5] or [6].

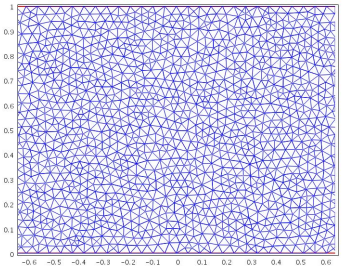


Fig. 7. Adapted triangular grid of $\hat{\Omega}$ (part of) - 9062 degrees of freedom

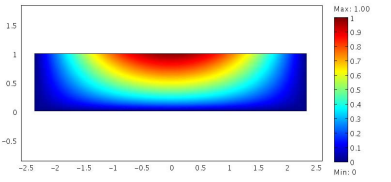


Fig. 8. Numerical solution (isolines) - 9062 degrees of freedom

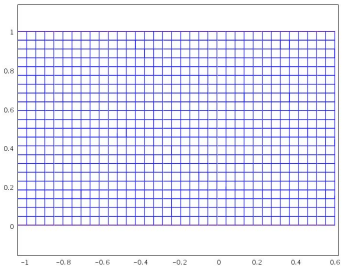


Fig. 9. Structured grid of rectangles (22×106) of $\hat{\Omega}$ (part of) - 2323 degrees of freedom

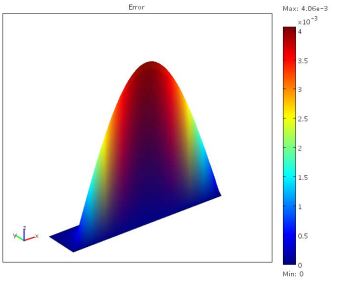


Fig. 10. Error - 2323 degrees of freedom

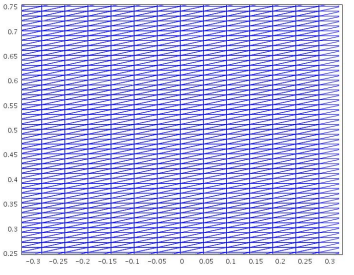


Fig. 11. Structured grid of triangles of $\hat{\Omega}$ (part of) - 10201 degrees of freedom

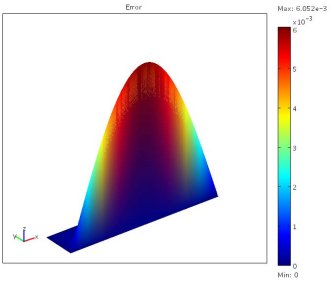


Fig. 12. Error - 10201 degrees of freedom

Figures 13-14 show the solution using an equidistant quadrilateral FVM grid of size 60×80 (see also Table 3).

Table 2. FEM results of polar coordinates

Method	Grid	Maximal error	degrees of freedom	Figure
FEM	adapted, unstructured	$2.98e-02$	9062	5-8
FEM	rectangular, structured	$4.06e-03$	2323	9-10

Table 3. FVM results of polar coordinates

Method	Grid	Maximal error	degrees of freedom	Figure
FVM	rectangular, structured	$5.6e-03$	4800	13-14
FVM	rectangular, structured	$4.6e-03$	8000	15-16

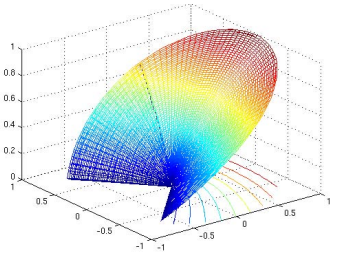


Fig. 13. FVM solution on a structured grid of rectangles (60×80) - 4800 degrees of freedom

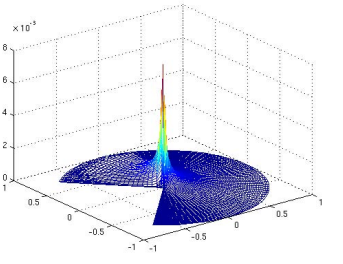


Fig. 14. Error of the FVM solution - 4800 degrees of freedom

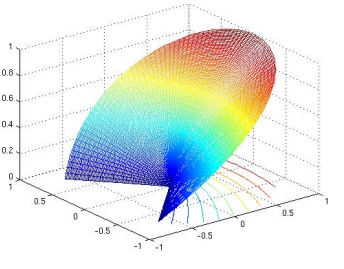


Fig. 15. FVM solution on a structured grid of rectangles (80×100)- 8000 degrees of freedom

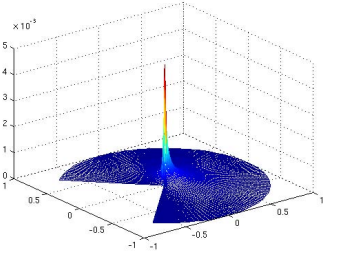


Fig. 16. Error of the FVM solution - 8000 degrees of freedom

With

$$-[\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial \hat{u}}{\partial r}+\frac{1}{r^2}\frac{\partial^2 \hat{u}}{\partial \phi^2}]=0 \quad \text{in } \hat{\Omega}, \quad \hat{u}(r,\phi)=\cos(\alpha\phi) \quad \text{on } \hat{\Gamma}=\partial\hat{\Omega}\,, \tag{20}$$

where

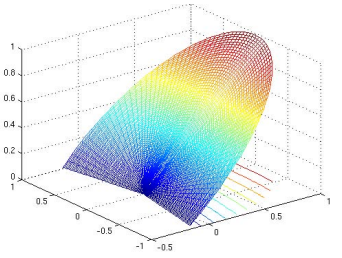


Fig. 17. FVM solution on Ω - 4800 degrees of freedom

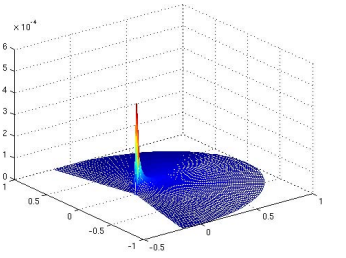


Fig. 18. Error of the FVM solution on Ω - 4800 degrees of freedom

$$\hat{\Omega} = \{(r, \phi) \in [0, 1] \times [-\frac{\pi}{2\alpha}, \frac{\pi}{2\alpha}]\}$$

is a circle sector including a singular boundary edge $\{(0, \phi),\ \phi \in [-\frac{\pi}{2\alpha}, \frac{\pi}{2\alpha}]\}$, i.e., a blunt angle and a concave region, we can generalize the test problem discussed above, where $\alpha \in]0, 1[$. We find with

$$\hat{u}(r, \phi) = r^{\alpha} \cos(\alpha\phi)$$

the exact analytic solution of problem (20). Figures 17, 18 show the FVM solution of problem (20) compared to the exact solution for $\alpha = \frac{9}{10}$.

The results show the decreased influence of the singular point (edge) with the outer angle of $\frac{10}{9}\pi$ to the error. Compared to the above discussed case ($\alpha = \frac{2}{3}$, outer angle $\frac{3}{2}\pi$ at the singular point), we observe an error reduction from $5.6e{-}03$ to $3.5e - 04$ using the same grid size.

4 Conclusion

Based on an appropriate coordinate transformation weak problem formulations and flux balances in curvelinear coordinate systems are constructed. Numerical tests prove the efficiency of the use of structured grids in FEM and FVM for appropriate geometries of integration regions of boundary value problems. It could be shown that structured rectangular FEM and FVM grids are optimal with respect to the error, the computational load and the storage requirements. The results for polar coordinates are also valid for cylindrical coordinates.

Acknowledgments. Especially in using COMSOL multiphysics I have to thank my colleague Thomas Slawig. The work is partial supported by the PROFIT project of the german government.

References

1. Fletcher, C.A.J.: Computational Techniques for Fluid Dynamics, vol. 1/2. Springer, Heidelberg (1991)
2. COMSOL multiphysics 3.3, users manual, reference guide, Göttingen (2006)
3. Bärwolff, G.: Numerik für Ingenieure Physiker und Informatiker, Elsevier - Spektrum Akademischer Verlag, München (2007)
4. Großmann, Chr., Roos, H.-G.: Numerik partieller Differentialgleichungen, B.G. Teubner, Leipzig (1994)
5. Smith, G.D.: Numerical solution of partial differential equations. Clarendon Press, Oxford (1985)
6. Roache, P.J.: Computational Fluid Dynamics. Hermosa Publishers, Albuquerque (1976)

Digital Simulation for Micro Assembly Arranged at Rectangular Pattern in Micro Factory

Murali Subramaniyam[1], Sangho Park[2], Sung-il Choi[3], Seokho Jang[4], and Joon-Yub Song[5]

[1] BK21 Mechatronics Group of Chungnam National University, murali@cnu.ac.kr
[2] BK21 Mechatronics Group of Chungnam National University, spark@cnu.ac.kr
[3] Graduate School of Chungnam National University, scadcam@paran.com
[4] BK21 Mechatronics Group of Chungnam National University, jegal3@cnu.ac.kr
[5] Korea Institute of Machinery & Materials, sjy658@kimm.re.kr

Abstract. In recent years, Micro Assembly has been identified as a critical technology in the promising Micro Systems Technology (MST). Most of the researchers have been working on miniaturization of parts and assembly and this paper describes the micro assembly with simulation in the digital environment. Miniaturization of machine tool leads to enormous saving in energy, space, time and resources in micro assembly, Micro Lens Module Assembly taken as an example to describe the digital simulation which arranged in rectangular pattern. In order to construct the micro factory, ultra small machine tools were arranged in required process sequence with resources and the digital simulation has been carried out for the micro factory using 3D CAD system, recyclable estimation also done with the digital simulation of the micro factory.

Keywords: Micro factory, micro machine, digital simulation.

1 Introduction

Micro factory refers to a small-size production system dedicated for the manufacture of very small parts requiring very high-precision, high throughput and low production costs and miniaturization of machine tool leads to enormous saving in energy, space, time and resources. In recent years, the digital manufacturing is the application of tools and a processing medium to the transformation of raw material into final products. In today's hyper-competitive environment, manufacturing organizations continually search for, develop, and fund projects that will reduce manufacturing costs and increase productivity. However, these traditional approaches are quickly reaching diminishing returns in their ability to reduce costs and improve productivity, a new approach is needed and Digital Manufacturing is the new approach, it has the

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 486–494, 2007.
© Springer-Verlag Berlin Heidelberg 2007

ability to virtually design, optimize, simulate and then execute a manufacturing process with a high degree of predictability.

It is also has the ability to closely couple the product design function to the process design function. Digital Manufacturing is a combination of open software and manufacturing methods that transforms production processes and business initiatives. It provides a solution that links all manufacturing disciplines including manufacturing process design, process simulation/engineering and production managements. It results in companies not only getting much more from their production operations, but more from product design investments and supply chain partnership. It is an essential component of PLM initiatives because it closes the gap between product design and product delivery by managing both the design and execution of manufacturing processes. The goal of the digital manufacturing is to provide the manufacturing community with solutions to create, validate, monitor and control agile [1]. The micro factory digital simulation for micro assembly is a challenge as well as opportunities to engineers and scientists, to enable drastic miniaturization of industrial products.

This paper gives the digital simulation for the micro parts assembly arranged at rectangular pattern in a micro factory based on Micro System Technology (MST). For the digital materialization of the new paradigm the components in the micro factory has been modeled using 3D CAD system. The modeling chapter includes resource modeling, micro assembly machine modeling, product modeling and process modeling. The micro lens module assembly taken as an example which has been modeled in our lab and analyzed with MST application module, the micro factory model has been modeled digitally using 3D CAD system and which was simulated with the assistance of UML Language (Unified Modeling Language) with object oriented logical analysis [2].

2 Building Micro Factory with Micro Assembly Machines

2.1 Micro Assembly Machine

With high degree of current micro-fabrication technology, researches aiming for the materialization of micro machines are actively carried out world wide [3], [4], [5], [6],

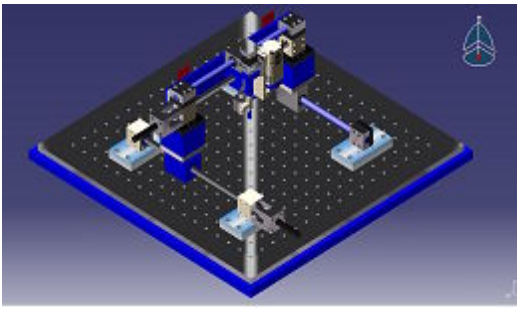


Fig. 1. Micro Part Assembly Machine

[7]. The assembly machine was modeled using 3D CAD system as shown in Fig.1, this micro assembly machine used to assemble the micro scaled parts in digital environment.

2.2 Building Digital Micro Factory

2.2.1 Product Modeling

Product modeling is considered as the preliminary design process. Micro Parts Assembly considered as a Product Model and it was modeled by 3D CAD system. Fig.2 shows the 3D product model of the micro lens assembly, which was assembled at micro assembly machine.

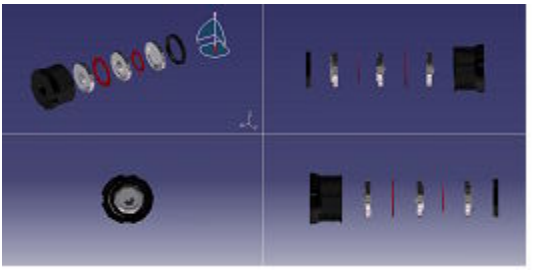


Fig. 2. Product Modeling

2.2.2 Resource Modeling

The lens module assembly system assembles the assembly, as we mentioned earlier, where lens module is taken as assembly. This lens module assembly system mainly is classified into different types of instrument as Y axis feed instrument, X-Z axis pickup head and actuator supporter. Depending on the appliance requirement it operates, when kinetic factor is granted. The micro engineering machine tool model and the instrument used in this system were modeled through the 3D CAD system as shown in Fig.3.

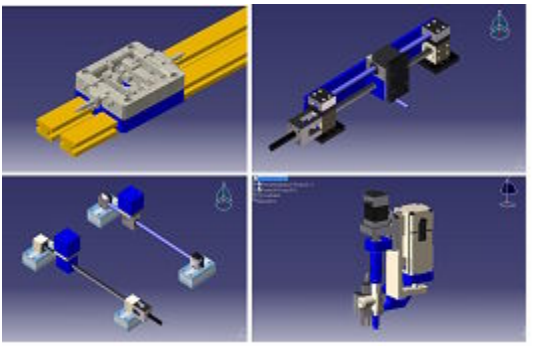


Fig. 3. Modules of the resources

2.2.3 Process Modeling

The co-ordinated set of actions that produces an output, either individual part or assembly called process. The micro engineering machine tool model of the cluster type lens module assembly system is a co-ordinated set of actions which was modeled from resource modeling. The process modeling contains the collection of micro machine tools, LM guide and the instrument tray. The instrument tray used to load the parts to the corresponding machines. The transferable table moves from one place to another place with the guidelines of LM guide. Fig. 4 shows the model of LM guide and tray, which also went through 3D CAD system.

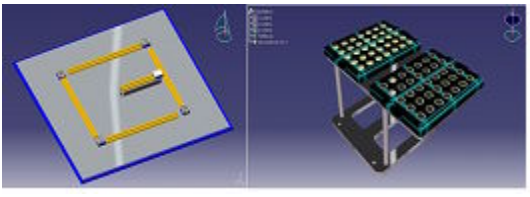


Fig. 4. 3D Model of Tray and LM guide

In order to perform process analysis to get visible and analytical execution of processes, the digital simulation concept was applied over the micro factory system by considering element, class, group, sub-model, simple logic / geometries / flow / consequences / manufacturing variables.

2.3 Building Digital Micro Factory Process Model

The micro factory assembly system was arranged in the micro parts assembly process sequences. The assembly process was built with the sequence as holder, lens3, spacer B, lens 2, spacer A, lens 1 and shield. Fig.1 refers the product model of the lens assembly module and its parts. Fig.5 shows the model and process sequences for lens assembly module in the digital environment.

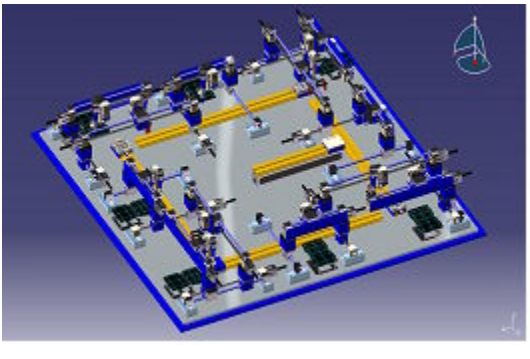


Fig. 5. Digital Micro Factory model and process

The instrument tray which is work holding device holds the parts, each assembly machine as tray with parts like holder, shield, lens, etc., the table which is transferable one, used to transfer the parts from one assembly machine to next assembly machine in the lens assembly module. The Unified Modeling Language (UML) is a non-proprietary specification language for object modeling. UML is a general-purpose modeling language that includes a standardized graphical notation used to create an abstract model of a system. An object modeling is a standardized set of symbols and ways of arranging them to model (part of) an object oriented software design or system design. This UML language helped us to arrange the micro engineering machines in micro factory environment.

Fig.5 shows the arrangement of micro engineering machines with resources models for lens assembly module system. The micro factory was obtained by arranging the resources like tray, transferable table, LM guide and micro machines. The assembly process sequence was arranged in rectangular pattern as shown in the Fig.5. The digital simulation was carried out over the micro factory for the lens assembly module, by allocating micro engineering machines in the resource with organizational characters of them as like actual operation.

With the aid of digital manufacturing technology on the micro factory process, based on process flow of PERT chart the complete assembly process was made as one cycle [2]. The timely analysis was executed over the lens assembly module with 67 simultaneous simulations. The time factor, which is time required for the rectangular movement of the process per road distance and which is same for simulation of the micro engineering machines. By setting time factor for each assembly process as 10mm/5s, simulation of the assembly system was carried out.

2.4 Recycling of Micro Assembly Machines

In order to avoid the error in the real manufacturing, prior simulation has been executed. This helps us to eliminate the errors before doing the real operation in the micro factory with the arrangement of micro engineering machines. Errors like conflicts and interferences are expected to happen during the real manufacturing. To find those errors the test has been made over the micro factory with the arrangement

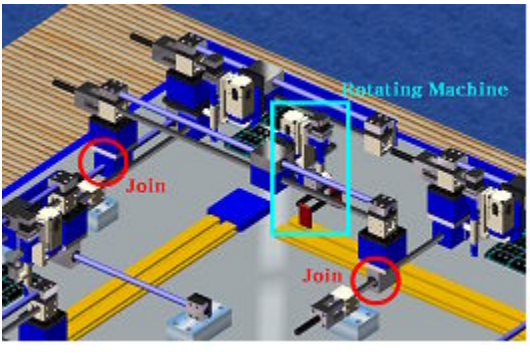


Fig. 6. Recycling of micro machine

of micro engineering machines and we found few corrections and modifications were needed in the road distance of actuator and pickup head.

The process arrangement of the system has been recycled with the correction of road distance. The rotating machine has the capability to share the submotor for rotation and for axis guidelines in the transfer line. The rotating machine used to transfer the parts from one transferable table to next when the absences of LM guide as shown in the Fig.6. The join represents the connection of motor axis and guide axis. The rotating machines use the motor axis as guide axis. The presences of LM guide lead the table travel on it and transfer the parts from one machine to another machine.

3 Simulation

The act of simulating something generally entails representing certain key characteristics or behaviors of a selected physical or abstract system. Simulation is an important feature in engineering systems or any system that involves many processes. The micro factory process simulation would play a major role in the form of increasing the manufacturing efficiency and would also be possible to design the optimized process line [3].

3.1 Simulation of Micro Assembly Machine Process Arrangement

As we mentioned earlier, the simulation is an important feature in engineering systems or any system involves many processes. Our micro lens module assembly involves many processes and arrangement, in the Fig. 7 the CAD model of the processes and its arrangement are shown. As we mentioned earlier the Lens Module assembly contains holder, lenses and spacers. The assembly processes of the example product were arranged in rectangular pattern/shape. The assembly process has been built with the sequence as holder, lens3, spacer B, lens 2, spacer A, lens 1 and ends with shield. According to this order of assembly the micro engineering machines were

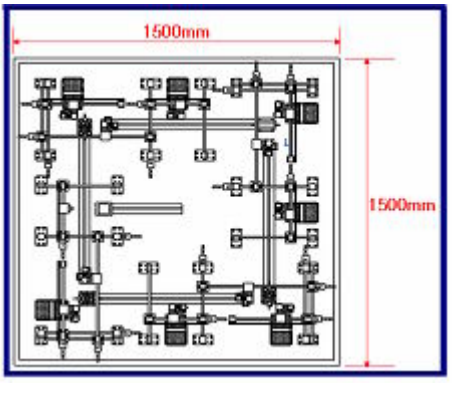


Fig. 7. Rectangular pattern arrangement of micro factory

arranged in the micro factory. The instrument tray acting as a work holding device for each assembly machines, it holds the parts required in the assembly machine. The transferable table acting as a material handling device, it moves the assembled parts from one machine to another machine over the LM guide.

The LM guide acting as a guide for the movement of transferable table from one assembly machine to another assembly machine and the arrangement of the LM guide shown in the Fig. 7. When the absence of LM guide, it is not possible to move the transferable table from one assembly machine to another assembly machine, to avoid this issue, the rotary machine was employed; it leads transformation of table from one table to another table. The rotating machine is capable to share the submotor for the rotating axis and the guide axis. As per the requirement of the assembly process, the assembly machines were arranged in rectangular pattern / shape with LM guide and the micro factory model has been built as shown in Fig.8.

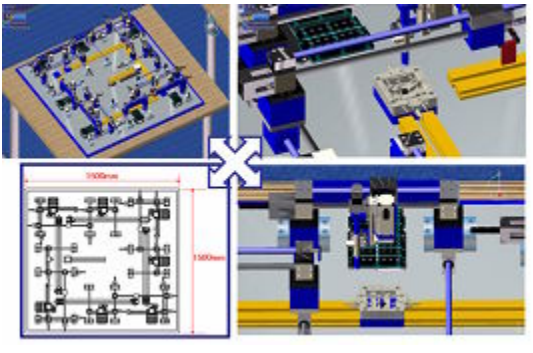


Fig. 8. Micro factory model

3.2 Process Simulation Result

As we mentioned earlier, the Fig. 7 shows the process arrangement of the micro factory in rectangular shape (pattern). The micro factory was built with the size of 1500 X 1500 in millimeter (length and breadth). The total processing time is refers, the time required to complete the one cycle of operation; it means the time required to assemble the lens module using micro assembly machine in a micro factory environment digitally. The assembly processes contains many processes and each processes takes its own processing time, using the individual processes time the total processing time was calculated. The total processing time calculation as follows:

Total process time = Total time + (Rotating x 4) + (Move LM Guide x 4) = 314.03 sec.

Table. 1 shows the simulation time required in the each assembly process and total processing time. The total process time has been calculated by adding the total time, material handling time and rotating time of processes in the micro factory. Total time is cumulative of assembly time required in the each assembly machines. The material handling time, time required to move the parts from one table to another table via LM

Table 1. Simulation time of micro factory

Micro machine / Lens module	X Axis pickup head	Z Axis picker	Y Axis feed mechanism
Holder	9.64s	2.059s	11.89s
Lens_3	4.52s	2.059s	7.17s
Spacer_B	9.64s	2.059s	11.89s
Lens_2	4.52s	2.059s	7.17s
Spacer_A	9.64s	2.059s	11.89s
Lens_1	4.52s	2.059s	7.17s
Shield	9.64s	2.059s	11.89s
Total time	52.1s	14.416s	69.07s
Total time	135.58s		
Rotating Robot	1.86s	9.37s	4.45s
LM Guide	28.94s		
Total Process	314.03s (5.234min)		

guide, it has four movements in LM guide. The rotating time when the absence of LM guide parts are transferred through the rotating machine.

4 Conclusion

This paper presents a digital environment for micro parts assembly with the lens module assembly as an example. The micro assembly, resources and processes for micro factory has been modeled using 3D CAD system, the process has been simulated realistically with the resources and the lens module assembly, it was manipulated efficiently for assembly evaluation. The prior simulation was executed over the micro factory with the arrangement of micro engineering machines. The correction and modification were made in the road distance and actuator pickup head. The process arrangement of the system was recycled with the correction. All signs have indicated that the dominant technologies in the new Millennium will be micro systems technology (MST) and nanotechnology (NT). Both these technologies will destine ultimate miniaturization of devices and engineering systems that will produce the dream products that will significantly improve human well-being and living standards. Active R&D in automated micro and nano-scaled assemblies offer great opportunities to engineers and scientists in this new Century.

Acknowledgements. This study is one of the Next-Generation Microfactory System Program. The authors are grateful for the support from the Ministry of Commerce, Industry and Energy, Korea and this work is (partially) supported by the second stage of Brain Korea 21 program.

References

1. Brown, R.G.: Driving Digital Manufacturing to Reality. In: Proceedings of the Winter Simulation Conference, pp. 224–228 (2000)
2. Jung, Y.S.: Assemblability Evaluation with Micro-factory Digital Simulation, Master Thesis, Chungnam National University (2006)
3. Park, J.K.: Technical Trends of Micro Factory. Journal of the Korean Society of Precision Engineering 19(10), 7–14
4. Kang, J.H.: Development of Micro factory System for Future Industry. Journal of the Korean Society of Precision Engineering 19(10), 15–22 (2002)
5. Jayaram, S., Jayaram, U.: VADE- A Virtual Assembly Design Environment, IEEE CGA, pp. 44–50 (1999)
6. Hsu, T.-R.: Micro Assembly - A technology on the frontier of new industrial automation. In: A keynote speech delivered at the 8th International Conference on Automation Technology, Taichung, Taiwan (May 5, 2005)
7. Choi, Y.H., Park, H.M., Jang, S.H., Park, J.K.: Structural Configuration Creating Method for Reconfigurable Micro Machine Tools. In: Proceedings of the Korean Society of Machine Engineers Spring Conference, pp. 123–128 (2006)

A New Quantized Input RLS, QI-RLS, Algorithm

A. Amiri[1,2], M. Fathy[2], M. Amintoosi[2,3], and H. Sadoghi[3]

[1] Islamic Azad University-Zanjan Branch
[2] Faculty of Computer Engineering, Iran University of Science and Technology
[3] Faculty of Engineering, Tarbiat Moallem University of Sabzevar, Iran
a_Amiri@comp.iust.ac.ir, mahFathy@iust.ac.ir,
mAmintoosi@iust.ac.ir

Abstract. Several modified RLS algorithms are studied in order to improve the rate of convergence, increase the tracking performance and reduce the computational cost of the regular RLS algorithm. . In this paper a new quantized input RLS, QI-RLS algorithm is introduced. The proposed algorithm is a modification of an existing method, namely, CRLS, and uses a new quantization function for clipping the input signal. We showed mathematically the convergence of the QI-RLS filter weights to the optimum Wiener filter weights. Also, we proved that the proposed algorithm has better tracking than the conventional RLS algorithm. We discuss the conditions which one have to consider so that he can get better performance of QI-RLS against the CRLS and standard RLS algorithms. The results of simulations confirm the presented analysis.

Keywords: Adaptive Filter, Recursive Least Square (RLS), Weiner Optimum Weights, Tracking.

1 Introduction

An adaptive filter is a filter which self-adjusts its transfer function according to an optimizing algorithm. Because of the complexity of the optimizing algorithms, most adaptive filters are digital filters that perform digital signal processing and adapt their performance based on the input signal. The Recursive Least Square (RLS) and the Least Mean Square (LMS) are two famous adaptive filtering algorithms [5]. They have attained its popularity due to a broad range of useful applications in such diverse areas as communications, radar, sonar, seismology, navigation and control systems, and biomedical electronics.

The optimization of convergence speed and tracking performance are open problems in adaptive filter theory. Fast convergence of the RLS has given rise to the development of the algorithms based on it [6,9,10,12].

The works reported in [1,2,3,7,8,14] have been done for increasing the real-time performance of the LMS algorithm using the sign of the input data and/or error during updating the filter weights. In the clipped RLS algorithm [11], the input signal is quantized into three levels of -1, 0, +1. They discussed the convergence and the computational complexity of their own CRLS algorithm.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, pp. 495 – 506, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In the three levels clipping method the small domain signal is assumed as noise and it is obvious that it causes a lot of lost in input information in signals with low SNR[1] . In the proposed new clipped RLS, QI-RLS, algorithm the input has been clipped, such that we save more information and in the same time, the convergence and tracking performance of the proposed algorithm, are remained. The mathematical proofs and simulation results shows the better performance of the proposed method against Standard RLS and CRLS.

The variants of RLS are discussed in Section 2. The proposed new algorithm, which is a modification of the aforementioned algorithm, appears in Section 3. Section 4 deals with computer simulation issues. The final section presents conclusion and summarizes the main findings.

2 RLS Algorithm

In this section we review the Standard RLS and the CRLS Algorithm [11] which is the foundation of our proposed algorithm.

2.1 Standard RLS Algorithm

RLS algorithm has been explained in many literatures such as [5, 13]. In this section we review briefly this algorithm. The RLS predictor algorithm has been studied in [11] as:

$$W_{n+1} = W_n + R_n^{-1} X_n e_n \tag{1}$$

Where:

$$R_n^{-1} = \lambda R_{n-1}^{-1} - \frac{\lambda^{-2} R_{n-1}^{-1} X_n X_n^T (R_{n-1}^{-1})^T}{1 + \lambda^{-1} X_n^T R_{n-1}^{-1} X_n} \tag{2}$$

$$e_k = d_k - X_k^T W_k \tag{3}$$

$W_k = [w_k(1), w_k(2), ..., w_k(N)]^T$ is the weight vector of the predictor, $X_k = [x_k(1), x_k(2), ..., x_k(N)]$ is the vector of the input data sequence, which is assumed to be a stationary random process, N is the number of filter tapes, e_k is the estimation error and d_k is the desired response.

2.2 Clipped RLS Algorithm

Sadoghi et al. [11] have proposed Clipped-RLS as a variation of the standard RLS algorithm. They quantized the input signal with msgn(.) into a three-level signal. The msgn(.) function is described in equation 4 and is shown in Figure 1:

[1] Signal to Noise Ratio.

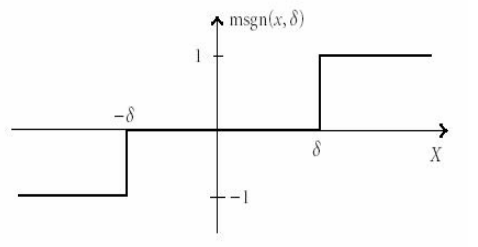


Fig. 1. Quantization scheme for the CRLS algorithm proposed in[11]

$$m\text{sgn}(x,\delta)=\begin{cases}1 & x>\delta\\ 0 & -\delta\le x\le\delta\\ -1 & x<\delta\end{cases} \tag{4}$$

According to msgn(.), the estimated input signal, $\hat{X}_n$ was replaced with X_n in Equations (2), (3). They also discussed the convergence and the computational complexity of their own CRLS algorithm.

In the next section we will explain the proposed new Quantized Input RLS (QI-RLS) and will discuss its performance, convergence and tracking.

3 The Proposed QI-RLS Algorithm

As we have seen in the previous section, the CRLS filter quantizes the input signal to three levels (Figure 1). Although it has been mentioned in [11] that it reduces the noise effect, but it is obvious that for inputs in range $[-\delta,+\delta]$, we have a lot of lost in input information. Because for that range, whole input assumed as noise. In the proposed new quantized input RLS, QI-RLS algorithm the input has been clipped such that we save all of the information for inputs in range $[-\delta,+\delta]$. Equation 4 shows our new clipping function that we named it $tgn(x,\delta)$ and is demonstrated in figure 2:

$$tgn(x,\delta)=\begin{cases}1 & x>\delta\\ x & -\delta\le x\le\delta\\ -1 & x<\delta\end{cases} \tag{5}$$

The adaptation equation can be written as:

$$W_{n+1}=W_n+R_n^{-1}X_n e_n \tag{6}$$

$$R_n^{-1}=\lambda R_{n-1}^{-1}-\frac{\lambda^{-2}R_{n-1}^{-1}\hat{X}_n\hat{X}_n^T(R_{n-1}^{-1})^T}{1+\lambda^{-1}\hat{X}_n^T R_{n-1}^{-1}\hat{X}_n} \tag{7}$$

$$e_k=d_k-X_k^T W_k \tag{8}$$

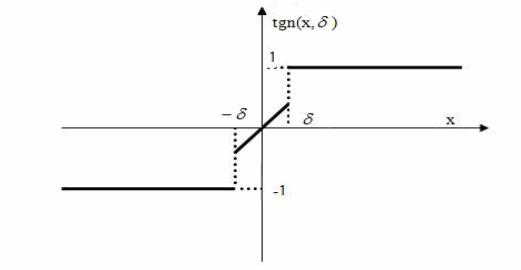


Fig. 2. Quantization scheme for the proposed clipping function, QI-RLS

Where $\hat{X}_n$ is the modified quantized input signal vector whose i^{th} component is: $\hat{x}_n(i) = tgn(x_n(i), \delta)$.

In the following subsections we explain the mathematical proof of the convergence and computational complexity reduction of the proposed method.

3.1 Convergence of QI-RLS

It is usual in adaptive filter literatures to prove the convergence of the filter weights to Weiner optimum weights. Theorem 1 proves the convergence of QI-RLS weights to Weiner weights.

Theorem 1. if the QI-RLS weights, W_n, is described by the recursive equation 6 and W_o is the Weiner optimum weight ,then W_n converges to W_o.

Proof:

We rewrite Equation (6) again: $W_{k+1} = W_k + R_k^{-1}\hat{X}_k e_k$

Where error is as follows:

$$e_k = d_k - X_k^T W$$

For convergence proof it is sufficient to show that: $\lim_{k\to\infty} E\{W_{k+1}\} = W_o$

Substituting e_k to Equation 6 yields: $W_{k+1} = W_k + R_k^{-1}(\hat{X}_k d_k - \hat{X}_k X_k^T W_k)$

Regarding the expectation of the above equation and Lemma 1 we have:

$$E\{W_{k+1}\} = E\{W_k\} + E\{R_k^{-1}\}(E\{\hat{X}_k X_k^T\}E\{W_k\})$$

$$= E\{W_k\} + E\{R_k^{-1}\}(\frac{\alpha'}{\sigma_x}E\{d_k X_k\} - \frac{\alpha'}{\sigma_x}E\{X_k^T X_k\}E\{W_k\})$$

$$\Rightarrow E\{W_{k+1}\} = E\{W_k\} + E\{R_k^{-1}\}\frac{\alpha'}{\sigma_x}(P - R.E\{W_k\}) \quad (9)$$

Where:

$$\alpha' = \sqrt{2/\pi}(1-\delta)\exp(-\delta^2/2\sigma_v^2) + \sigma_v erf(\delta/\sqrt{2}\,\sigma_v)$$

Here, we have to compute $E\{R_k^{-1}\}$.

By considering Lemma 2, $E\{R_k^{-1}\}=R^{-1}(1-\lambda)$, Equation (9) can be rewritten as follows:

$$E\{W_{k+1}\}=E\{W_k\}+R^{-1}(1-\lambda)\frac{\alpha'}{\sigma_x}(P-R.E\{W_k\})$$

$$\Rightarrow E\{W_{k+1}\}=E\{W_k\}(1-(1-\lambda)R^{-1}R\frac{\alpha'}{\sigma_x})+(1-\lambda)R^{-1}.P.\frac{\alpha'}{\sigma_x}$$

The Weiner solution is $W_o=R^{-1}.P$. Hence:

$$E\{W_{k+1}\}=E\{W_k\}(1-(1-\lambda)\frac{\alpha'}{\sigma_x})+(1-\lambda)\frac{\alpha'}{\sigma_x}W_o \tag{10}$$

Also, we know that:

$$\underset{k\to\infty}{Lim}E\{W_{k+1}\}=\underset{k\to\infty}{Lim}E\{W_k\}$$

By taking the limit of Equation (10) and after some simplification, we have:

$$\underset{k\to\infty}{Lim}E\{W_{k+1}\}\left(1-(1-(1-\lambda)\frac{\alpha'}{\sigma_x})\right)=(1-\lambda)\frac{\alpha'}{\sigma_x}W_o$$

$$\Rightarrow \underset{k\to\infty}{Lim}E\{W_{k+1}\}=W_o\ .$$

Hence, the proof is complete. □

3.2 Tracking Performance of QI-RLS

Tracking in filter theory, means tracking of the filter weights. According to [4,5,11], tracking is a steady-state phenomenon that is different from the convergence, which is a transient phenomenon. In general, convergence and tracking are two different properties. That is, if an algorithm has good convergence, its tracking ability is not necessarily fast and vice versa. In the tracking phase, a reasonable assumption is that the optimum weights vary according to a first-order Markov process [5], and the filter must track these weights. The following relation shows the variation of the filter's optimum weights:

$$\begin{aligned} W_{n+1}^{*} &= aW_n^{*}+\omega_n \\ d_n &= W_n^{*T}X_n+\upsilon_n \end{aligned} \tag{11}$$

Where a is a constant and ω_n is the process noise vector in the n[th] step, which has zero mean, and υ_n is the measurement noise, which is assumed to be white Gaussian with zero mean and variance σ_v^2.

Missadjustment criterion as shown in Equation (12) is a usable measure for tacking performance [5]:

$$M=\frac{E\{\left|\omega_n^T X_n\right|^2\}}{E\{\left|\upsilon_n\right|^2\}} \tag{12}$$

In the following theorem we will show the relation between our QI-RLS and the conventional RLS according to this criterion.

Theorem 2. Let M_{QI-RLS} and M_{RLS} are QI-RLS and conventional RLS miss adjustment criterion, respectively, then:

$$M_{QI-RLS} = \left(\frac{\alpha'}{\sigma_x}\right)^2 M_{RLS} \tag{13}$$

Where:

$$\alpha' = \sqrt{2/\pi}(1-\delta)\exp(-\delta^2/2\sigma_x^2) + \sigma_x erf(\delta/\sqrt{2}\,\sigma_x)\,.$$

Proof:

$$E\{\left|\omega_n^T \hat{X}_n\right|^2\} = E\{\omega_n^T \hat{X}_n \hat{X}_n^T \omega_n\}$$

With suppose the independency between ω_n and $\hat{X}_n$. By using Lemma 1 we have:

$$E\{\left|\omega_n^T \hat{X}_n\right|^2\} = E\{\omega_n^T \hat{X}_n \hat{X}_n^T \omega_n\} = E\{\omega_n^T \hat{X}_n\} E\{\hat{X}_n^T \omega_n\}$$

$$= \left(\frac{\alpha'}{\sigma_x} E\{\omega_n^T X_n\}\right)\left(\frac{\alpha'}{\sigma_x} E\{X_n^T \omega_n\}\right) = \left(\frac{\alpha'}{\sigma_x}\right)^2 E\{\left|\omega_n^T X_n\right|^2\}$$

By dividing the equation to $E\{|\upsilon_n|^2\}$:

$$\frac{E\{\left|\omega_n^T \tilde{X}_n\right|^2\}}{E\{|\upsilon_n|^2\}} = \left(\frac{\alpha'}{\sigma_x}\right)^2 \frac{E\{\left|\omega_n^T X_n\right|^2\}}{E\{|\upsilon_n|^2\}} \quad \Rightarrow M_{QI-RLS} = \left(\frac{\alpha'}{\sigma_x}\right)^2 M_{RLS}\,. \qquad \square$$

In an instance filter, the reduction of the missadjustment measure means the increase of tracking performance [5]. Hence, from theorem 2 it is obvious that if $\alpha' < \sigma_x$, tracking performance of QI-RLS is better than RLS. But it does not discuss conditions which one have to consider so that he can get better performance of QI-RLS. Theorem 3 describes the suitable parameter values.

Theorem 3. if $\delta = \sqrt{2}\sigma_x$ then Tracking performance of QI-RLS algorithm is better than conventional RLS for $\sigma_x > 0.513$. Where σ_x is the variance of input signal, and δ is the parameter of $tgn(x,\delta)$.

Proof: It is obvious that, if the missadjustment of QI-RLS is lower than the missadjustment of conventional RLS, then tracking performance of QI-RLS is better than RLS. Regarding to results of theorem 2, we have:

$$M_{NC-RLS} = \left(\frac{\alpha'}{\sigma_x}\right)^2 M_{RLS}$$

Where $\alpha' = \sqrt{2/\pi}(1-\delta)\exp(-\delta^2/2\sigma_v^2) + \sigma_v erf(\delta/\sqrt{2}\,\sigma_v)$

Now, it is sufficient to show that if $\delta = \sqrt{2}\sigma_x$ then $\alpha' < \sigma_x$, hence:

$$\alpha' = \sqrt{2/\pi}(1-\sqrt{2}\sigma_x)\exp(-(\sqrt{2}\sigma_x)^2/2\sigma_x^2) + \sigma_x erf(\sqrt{2}\sigma_x/\sqrt{2}\,\sigma_x) < \sigma_x$$

After some simplification we have:

$$\sigma_x > \frac{\sqrt{2/\pi}\exp(-1)}{1-erf(1)+2/\sqrt{\pi}\exp(-1)} \approx 0.513$$

So, considering $\sigma_x > 0.513$ and $\delta = \sqrt{2}\sigma_x$ implies $\alpha' < \sigma_x$, and the proof is complete. □

Following section shows some experimental results demonstrating the better performance of the QI-RLS against to some others.

4 Experimental Results: Predicting a Noisy Chirp Signal

As mentioned earlier on theorem 1, the proposed filter weights converges to Wiener optimum weights. The Wiener optimum weights are computed as follow:

$$W_o = R^{-1}P \tag{14}$$

Where: $R = E\{X_k X_k^T\}$ and $P = E\{d_k X_k\}$.

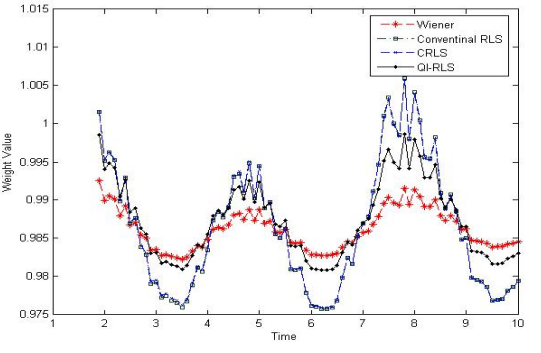


Fig. 3. Weight Tracking Comparison between Wiener filter, RLS, CRLS and the proposed QI-RLS

Figure 3 shows the weight tracking of our method, standard RLS and CRLS [5] for tracking a noisy chirp sinusoid signal, which was shown in Figure 4. As Equation (14) shows, for computing the Wiener Weights, all of data must be in hand. For comparison purpose, at every time, we used the incoming data until that time. The following pseudo code shows our used algorithm for computing Wiener Optimum weights[2].

```
X: Input Signal
Y: Desired Output where Ly=Length(Y)
N: Number of last input for prediction
```

[2] The original code is about to Rich Kozick.

```
For k= N: Ly
        x_k = X_(k-N+1), …, X_(k);
        P = P + Y_(k)*x_k;
        R = R + x_k*x_k^T;
        Weiner(k) =R^-1 * P;
End
OptWeights = R^-1 * P;    //Optimum Weiner Weights
```

As can be seen in Figure 3 the weights of the QI-RLS are closer to Wiener weights comparing to others.

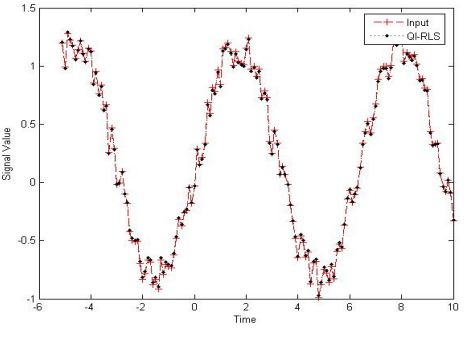


Fig. 4. Signal Tracking: Tracking a noisy chirp sinusoid signal by QI-RLS

Also we saw in Theorem 3, that the tracking performance of QI-RLS algorithm is better than conventional RLS, based on some conditions. The result of running our algorithm for 40 noisy signals, regarding the theorem 3 conditions, shows that the proposed method can produce better results comparing to some others.

The mean square error of an estimator $\hat{\theta}$ of a parameter θ represented by (15) is a widely used criterion for comparing estimator algorithms.

$$MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)^2] \tag{15}$$

Figure 5 shows the mean squared error of the proposed QI-RLS, conventional RLS and CRLS predicted signals with the original noisy signal. The original signal is considered as θ and the predicted signal as $\hat{\theta}$ in Equation (15).The algorithms were run on 40 random chirp sinusoidal signals and for every signal the error estimation are computed. As can be seen the related MSE of the proposed filter and the CRLS filter are lower than conventional RLS. Closing the result of the proposed method and the CRLS, in Fig 5, makes it hard to distinguish which of them is really better. Hence we have used another criterion in Figure 6 for better demonstration.

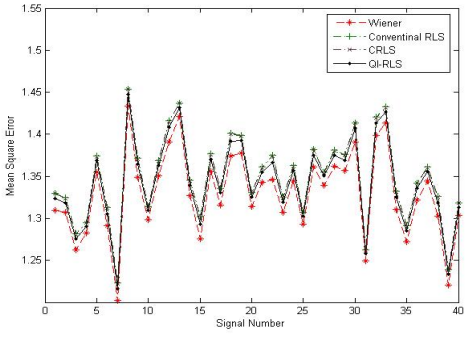


Fig. 5. Comparing MSE of the proposed QI-RLS, conventional RLS and CRLS predicted signals with the original noisy signal

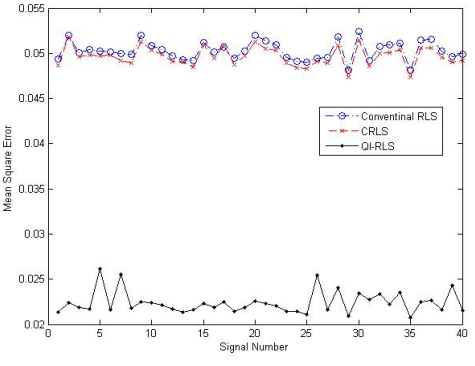


Fig. 6. Weight estimation error, MSE of the difference weight vector between proposed QI-RLS, conventional RLS, CRLS weights and Wiener optimum weights

5 Conclusion

Prediction is a major part of tracking algorithms. One of the most commonly used algorithms in prediction is the RLS algorithm. In this paper we proposed a new variant of the RLS, namely, the QI-RLS algorithm. This algorithm uses a quantization scheme which involves the threshold clipping of the input signal. Mathematical analysis shows the convergence of the filter weights to the optimum Wiener weights. Also we showed under which circumstances tracking performance of QI-RLS algorithm is better than conventional RLS. Experimental results on predicting a noisy chirp signal demonstrated the good performance of the proposed algorithm.

Acknowledgment. The authors wish to thank to Rich Kozick and St.Weiss for their free MATLAB codes.

References

1. Deivasigamani, L.: A fast clipped-data LMS algorithm. IEEE Trans. Acoustics, Speech, and Signal Processing 30(4), 648–649 (1982)
2. Eweda, E.: Analysis and design of a signed regressor LMS algorithm for stationary and nonstationary adaptive filtering with correlated Gaussian data. IEEE Trans. Circuits Syst. 37(11), 1367–1374 (1990)
3. Eweda, E.: Comparison of RLS, LMS, and sign algorithms for tracking randomly time-varying channels. IEEE Trans. Signal Processing 42(11), 2937–2944 (1994)
4. Feur, A., Weinstein, E.: Convergence analysis of LMS filters with uncorrelated Gaussian data. IEEE Trans. Acoust, Speech, Signal Processing 33, 222–230 (1985)
5. Haykin, S.: Adaptive Filter Theory. Prentice-Hall, Englewood Cliffs (1996)
6. Haykin, S., Sayed, A.H., Zeidler, J., Yee, P., Wei, P.: Tracking of linear time-variant systems. In: Proc. IEEE Military Communications Conference (MILCOM '95), San Diego, Calif, USA, vol. 2, pp. 602–606 (1995)
7. Kwong, C.: Dual sign algorithm for adaptive filtering. IEEE Trans. Commun. 34(12), 1272–1275 (1986)
8. Mathews, V., Cho, S.H.: Improved convergence analysis of stochastic gradient adaptive filters using the sign algorithm. IEEE Trans. Acoustics, Speech, and Signal Processing 35(4), 450–454 (1987)
9. Sadoghi Yazdi, H., Fathy, M., Lotfizad, M.: Vehicle tracking at traffic scene with modified RLS. In: Campilho, A., Kamel, M. (eds.) ICIAR 2004. LNCS, vol. 3212, pp. 623–632. Springer, Heidelberg (2004)
10. Sadoghi Yazdi, H., Lotfizad, M., Kabir, E., Fathy, M.: Application of trajectory learning in tracking vehicles in the tra.c scene. In: Proc. 9th Annual Computer Society of Iran Computer Conference(CSICC '04), vol. 1, Tehran, Iran, pp. 180–187 (2004)
11. Sadoghi Yazdi, H., Lotfizad, M., Kabir, E., Fathy, M.: Clipped input RLS Applied to Vehicle Tracking. EURASIP Journal on Applied Signal Processing 8, 1229–1234 (2005)
12. Vaseghi, S.: Advanced Signal Processing and Digital Noise Reduction. JohnWiley & Sons, New York (1996)
13. Widrow, B., Stearns, S.D.: Adaptive Signal Processing, Chs. 2-6. Prentice-Hall, Englewood Cliffs (1985)
14. Sethares, W.A., Mareels, I.M.Y., Anderson, B.D.O., Johnson C. R., J., Bitmead, R.R.: Excitation conditions for signed regressor least mean squares adaptation. IEEE Trans. Circuits Syst., 35(6), 613–624 (1988)
15. Eleftheriou, E., Falconer, D.: Tracking properties and steady-state performance of RLS adaptive filter algorithms. IEEE Trans. Acoustics, Speech, and Signal Processing 34(5), 1097–1110 (1986)

Appendix

Lemma 1. If two random variables u and v both have a Gaussian distribution $N(0,\sigma_u)$ and $N(0,\sigma_v)$ respectively and $E\{uv\}=\rho\sigma_u\sigma_v$, $\hat{v}=tgn(v,\sigma)$ then:

$$E\{u\hat{v}\}=\frac{\alpha'}{\sigma_v}E\{uv\}$$

Where $\alpha'=\sqrt{2/\pi}(1-\delta)\exp(-\delta^2/2\sigma_v^2)+\sigma_v erf(\delta/\sqrt{2}\,\sigma_v)$

Proof:

We define the random variable $z = \frac{u}{\sigma_u} + \rho\frac{v}{\sigma_v}$

Now we have:

$$E\{zv\} = E\left\{\left(\frac{u}{\sigma_u} - \rho\frac{v}{\sigma_v}\right)v\right\} = E\left\{\frac{uv}{\sigma_u}\right\} - E\left\{\frac{\rho v^2}{\sigma_v}\right\}$$

With regard to assumption of the theorem $E\{zv\} = \frac{\rho\sigma_u\sigma_v}{\sigma_u} - \frac{\rho\sigma_v^2}{\sigma_v} = 0$

Therefore z and v are uncorrelated. Here we show that z and $\hat{v}$ are uncorrelated too. We have to show $E\{z\hat{v}\} = 0$. At first we prove that $E\{\hat{v}\} = 0$.

$$E\{\hat{v}\} = \int_{-\infty}^{\infty}\hat{v}f(\hat{v})d\hat{v} = \int_{-\infty}^{\infty}\hat{v}\exp(\frac{-\hat{v}^2}{\sigma_{\hat{v}}^2})d\hat{v} = \int_{-\infty}^{-\delta} -\exp(\frac{-1}{\sigma_{\hat{v}}^2})d\hat{v} + \int_{-\delta}^{\delta} v\exp(\frac{-v^2}{\sigma_{\hat{v}}^2})dv + \int_{\delta}^{\infty}\exp(\frac{-1}{\sigma_{\hat{v}}^2})d\hat{v} = 0$$

$$E\{z\hat{v}\} = E\{z\}E\{\hat{v}\} = 0 \Rightarrow E\left\{\left(\frac{u}{\sigma_u} - \rho\frac{v}{\sigma_v}\right)\hat{v}\right\} = 0 \Rightarrow \frac{1}{\sigma_u}E\{u\hat{v}\} = \frac{\rho}{\sigma_v}E\{v\hat{v}\}$$

Therefore, we have:

$$E\{u\hat{v}\} = \rho\frac{\sigma_u}{\sigma_v}E\{v\hat{v}\} \tag{A-1}$$

On the other hand

$$v\hat{v} = v\times tgn(v,\delta) = \begin{cases} |v|, & |v| > \delta \\ v^2, & |v| \le \delta \end{cases}$$

$$E\{v\hat{v}\} = \int_{-\infty}^{\infty} v\hat{v}\frac{1}{\sqrt{2\pi}\delta_v}\exp(\frac{-v^2}{\sigma_v^2})dv$$

$$= \int_{-\infty}^{-\delta}|v|\frac{1}{\sqrt{2\pi}\delta_v}\exp(\frac{-v^2}{2\sigma_v^2})dv + \int_{-\delta}^{\delta}v^2\frac{1}{\sqrt{2\pi}\delta_v}\exp(\frac{-v^2}{2\sigma_{\hat{v}}^2})dv + \int_{\delta}^{\infty}|v|\frac{1}{\sqrt{2\pi}\delta_v}\exp(\frac{-v^2}{2\sigma_v^2})dv$$

$$\Rightarrow E\{v\hat{v}\} = \frac{2}{\sqrt{2\pi}\delta_v}\int_{+\delta}^{+\infty}v\exp(\frac{-v^2}{2\sigma_v^2})dv + \frac{1}{\sqrt{2\pi}\delta_v}\int_{-\delta}^{+\delta}v^2\exp(\frac{-v^2}{2\sigma_{\hat{v}}^2})dv$$

Here we calculate the above two terms:

$$\frac{2}{\sqrt{2\pi}\delta_v}\int_{+\delta}^{+\infty}v\exp(\frac{-v^2}{2\sigma_v^2})dv = \sqrt{\frac{2}{\pi}}\sigma_v\exp(\frac{-\delta^2}{2\sigma_v^2})$$

$$\frac{1}{\sqrt{2\pi}\delta_v}\int_{-\delta}^{+\delta}v^2\exp(\frac{-v^2}{2\sigma_{\hat{v}}^2})dv = \frac{1}{\sqrt{2\pi}\delta_v}\{-2\delta\exp(\frac{-\delta^2}{2\sigma_{\hat{v}}^2})\sigma_v^2 + \sigma_v^3 erf(\frac{\delta}{\sqrt{2}\sigma_v})\}$$

$$= -\delta\sqrt{\frac{2}{\pi}}\sigma_v\exp(-\frac{\delta^2}{2\sigma_v^2}) + \sigma_v^2 erf(\frac{\delta}{\sqrt{2}\sigma_v})$$

Substituting the 2,3 in 1, and after some simplifications we have:

$$E\{v\hat{v}\} = \sqrt{\frac{2}{\pi}}\sigma_v(1-\delta)\exp(\frac{-\delta^2}{2\sigma_v^2}) + \sigma_v^2 erf(\frac{\delta}{\sqrt{2}\sigma_v}) \quad \text{(A-2)}$$

Now we can compute $E\{u\hat{v}\}$.Regarding (A-1) and (A-2) we have:

$$E\{u\hat{v}\} = \frac{\rho\sigma_u}{\sigma_v}\left\{\sqrt{\frac{2}{\pi}}\sigma_v(1-\delta)\exp(\frac{-\delta^2}{2\sigma_v^2}) + \sigma_v^2 erf(\frac{\delta}{\sqrt{2}\sigma_v})\right\}$$

$$E\{u\hat{v}\} = \rho\sigma_u\sigma_v\left\{\frac{1}{\sigma_v}\sqrt{\frac{2}{\pi}}(1-\delta)\exp(\frac{-\delta^2}{2\sigma_v^2}) + erf(\frac{\delta}{\sqrt{2}\sigma_v})\right\} \Rightarrow$$

$$E\{u\hat{v}\} = \frac{1}{\sigma_v}E\{uv\}\left\{\sqrt{\frac{2}{\pi}}(1-\delta)\exp(\frac{-\delta^2}{2\sigma_v^2}) + \sigma_v erf(\frac{\delta}{\sqrt{2}\sigma_v})\right\}$$

Hence with defining α' as follows: $\alpha' = \sqrt{2/\pi}(1-\delta)\exp(-\delta^2/2\sigma_v^2) + \sigma_v erf(\delta/\sqrt{2}\,\sigma_v)$

So: $E\{u\hat{v}\} = \frac{\alpha'}{\sigma_v}E\{uv\}$. □

Lemma 2. if $R_k = \sum_{i=1}^{k}\lambda^{k-i}X_iX_i^T$ and $R = E\{X_iX_i^T\}$ then $E\{R_k^{-1}\} = R^{-1}(1-\lambda)$.

Proof:

The expectation of R_k will be: $E\{R_k\} = R_k = \sum_{i=1}^{k}\lambda^{k-i}E\{X_iX_i^T\}$

According to Eleftheriou and Falconer's theorem [15], $E\{R_k\} = R_k$ and, regarding to lemma assumption, $R = E\{X_iX_i^T\}$, we have:

$$E\{R_k\} = R_k = \sum_{i=1}^{k}\lambda^{k-i}R = (1+\lambda+\lambda^2+...+\lambda^{k-1})R$$

If $k \to \infty$ and $\lambda < 1$ then: $E(R_k) = R_k = {}^{1}\!/_{(1-\lambda)}R$

Taking the inverse of the above equation leads to: $E(R_k^{-1}) = R^{-1}(1-\lambda)$ and the proof is complete. □

Decentralized Replica Exchange Parallel Tempering: An Efficient Implementation of Parallel Tempering Using MPI and SPRNG

Yaohang Li[1], Michael Mascagni[2], and Andrey Gorin[3]

[1] Department of Computer Science
North Carolina A&T State University, Greensboro, NC 27411
yaohang@ncat.edu
[2] Department of Computer Science
Florida State University, Tallahassee, FL 32306
mascagni@cs.fsu.edu
[3] Division of Computer Science and Mathematics
Oak Ridge National Laboratory, Oak Ridge, TN 37831
agor@ornl.gov

Abstract. Parallel Tempering (PT), also known as Replica Exchange, is a powerful Markov Chain Monte Carlo sampling approach which aims at reducing the relaxation time in simulations of physical systems. In this paper, we present a novel implementation of PT, so-called decentralized replica exchange PT, using MPI and the Scalable Parallel Random Number Generators (SPRNG) libraries. By adjusting the replica exchange operations in the original PT algorithm, and taking advantage of the characteristics of pseudorandom number generators, this implementation minimizes the overhead caused by interprocessor communication in replica exchange in PT. This enables one to efficiently apply PT to large-scale massively parallel systems. The efficiency of this implementation has been demonstrated in the context of various benchmark energy functions, such as the high-dimensional Rosenbrock function, and a rugged funnel-like function.

Keywords: Parallel Tempering, Monte Carlo Methods, Parallel Programming.

1 Introduction

Parallel Tempering (PT), also known as Replica Exchange or the Multi-Markov Chain method, is a powerful Markov Chain Monte Carlo (MCMC) sampling scheme proposed by Geyer and Thompson [1], and Marinari and Parisi [2]. In parallel tempering, multiple independent replicas of a system are simulated simultaneously under different thermodynamic conditions, the differences defined by temperatures in most cases. Replicas at high temperature are generally capable of experiencing a larger volume of the phase space while those at low temperature are able to explore the "local detail" of the energy landscape. During the process of simulation, neighboring replicas are allowed to exchange configurations from time to time, subject to the acceptance criterion. By carefully setting up the temperature ladder and the number of

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 507 – 519, 2007.
© Springer-Verlag Berlin Heidelberg 2007

replicas, PT can reduce the relaxation time of the Monte Carlo simulations in the physical systems, and improve convergence to a global minimum. PT is ideal for complex physical systems that are characterized by rough energy landscapes. Successful PT applications include the simulation of biomolecules [3], determination of X-ray structures [4], polymers [5], and structure prediction in small proteins [6], [7].

Intuitively, PT simulation is a natural fit for parallel computing systems because multiple replicas are allowed to run simultaneously at different temperatures. Each replica simulation can be realized as an independent process running on its own CPU. However, replica exchange operations in PT can become computationally expensive for large-scale simulations, due to the number of replicas needed as well as the interprocessor communication overhead between replicas.

In this paper, we present our novel decentralized replica exchange parallel tempering implementation. Our implementation is based on the MPI and SPRNG (Scalable Parallel Random Number Generators) [8] libraries. Functions in the MPI library are used for necessary interprocessor communication in the parallel computing environment. The SPRNG library provides parameterized pseudorandom number generators to produce independent random number streams for parallel processes. By taking advantage of the determinism and reproducibility characteristics of pseudorandom number streams, distributed processes can come to a common decision without performing interprocessor communication. Moreover, temperature exchange instead of configuration exchange is used to reduce the amount of communication in replica exchange. To eliminate the additional global synchronization posed by temperature exchange, we extend the neighboring replica exchange in the original PT scheme to a more generalized random replica exchange. All these efforts lead to a decentralized implementation of replica exchange transitions in PT, and thus minimize the interprocessor communication overhead in parallel PT applications.

2 The Parallel Tempering Scheme

In a general, the PT algorithm using MCMC for local sampling works as follows. A composite system with N sets of replicas is constructed with one replica per temperature level, T_i. Multiple temperature levels form a temperature ladder. A state of the composite system is specified by $X = \{x_1, x_2, \ldots, x_N\}$, where x_i is the replica at temperature level i. The equilibrium distribution of the composite system, X, is,

$$\Pi(X) = \prod_{i=1}^{N} \frac{e^{-\beta_i E(x_i)}}{Z(T_i)}, \tag{1}$$

where $\beta_i = 1/T_i$, $E(x_i)$ is the energy function, and $Z(T_i) = \int e^{-\beta_i E(x_i)} dx_i$, is the partition function of the replica at T_i.

At each iteration step, t, the Markov chains can be realized with two types of transitions – the Metropolis transition and the replica transition:

1. Metropolis Transition: The Metropolis transition is employed for local Monte Carlo moves for the conformation at each temperature level. The transition probability only depends on the change of in the objective function, $E(x_i)$, where x_i is

the conformation at temperature level T_i. A new configuration x_i' is sampled from the proposal distribution $q_i(.|x_i)$. The Metropolis-Hastings ratio at temperature level T_i is calculated as:

$$w_{Local}(x_i \to x_i') = e^{-\beta_i \Delta_i E} = e^{-\beta_i (E(x_i')-E(x_i))}, \tag{2}$$

The new state is accepted with the probability $\min(1, w_{Local}(x_i \to x_i'))$. The detailed balance condition holds for each replica in Metropolis transition and therefore, it also holds for the composite system.

2. Replica Transition: The replica transition takes place with the probability θ and is used to exchange conformations at two neighboring temperature levels, i and $i+1$.

$$x_i \leftrightarrow x_{i+1}. \tag{3}$$

The exchange is accepted according to the Metropolis-Hastings criterion with probability

$$\begin{aligned} &P_{\text{Re}\,plica}(x_i \leftrightarrow x_{i+1}) = P(\{x_1,...,x_i,x_{i+1},...,x_N\} \mid \{x_1,...,x_{i+1},x_i,...,x_N\}) \\ &= \min(1, \frac{\Pi(\{x_1,...,x_{i+1},x_i,...,x_N\})}{\Pi(\{x_1,...,x_i,x_{i+1},...,x_N\})}) \\ &= \min(1, e^{-\beta_i E(x_{i+1})-\beta_{i+1}E(x_i)+\beta_{i+1}E(x_{i+1})+\beta_i E(x_i)}) \end{aligned}. \tag{4}$$

The relaxation rate [9] can be characterized by the ergodic measure via the so-called fluctuation metric,

$$\Omega(X^t) = \sum_{j=1}^{N} [\beta_j E(x_j^t) - \overline{\beta E(x^t)}]^2 / N, \tag{5}$$

where $\overline{\beta E(x^t)} = \sum_{k=1}^{N} \beta_k E(x_k{}^t)/N$ is the ergodic average at iteration step t. The replica transitions lead to an improvement of the relaxation rate of the overall simulation of the composite system. Using the definition of the replica exchange probability, the detailed balance equation can be obtained for replica transition.

$$\begin{aligned} &P(\{x_1,...,x_i,x_{i+1},...,x_N\} \mid \{x_1,...,x_{i+1},x_i,...,x_N\})\Pi(\{x_1,...,x_{i+1},x_i,...,x_N\}) \\ &= P(\{x_1,...,x_{i+1},x_i,...,x_N\} \mid \{x_1,...,x_i,x_{i+1},...,x_N\})\Pi(\{x_1,...,x_i,x_{i+1},...,x_N\}) \end{aligned} \tag{6}$$

Descriptive pseudo code of the PT algorithm follows.

```
Initialize N replica x_1, x_2, …, x_N and their corre-
sponding temperatures T_1, T_2, …, T_N
Initialize t ← 0
Repeat {
// Perform Metropolis Transition
for each replica i {
   Sample a point x_i' from q_i( . | x_i )
   Sample a uniform [0, 1) random variable U_M
```

```
    if U_M <= w_local(x_i → x_i') then x_i ← x_i'
    }
//Perform Replica Transition
Sample a uniform [0, 1) random variable U_R
if U_R <= θ then {
    Sample an integer variable i from U[1, N-1]
    Sample a uniform [0, 1) random variable U_S
    if  U_S <= P_Replica(x_i↔x_{i+1}) then
            x_i ↔ x_{i+1}
    }
Increment t
}
```

3 Decentralized Parallel Implementation

3.1 Pseudorandom Number Reproducibility for Global Process Synchronization

In PT algorithms, a common decision has to be made among multiple processes to determine whether the replica transition should occur. The common decision is based on a uniform [0, 1) random number. Instead of producing a uniform pseudorandom number and then broadcasting it to other processes, a clever implementation is to use a random number generator with the same parameters and seed, for the replica transition decision in each individual process. A pseudorandom number generator is deterministic and reproducible, i.e., with the same parameters and seed, the generator will always produce the identical random number stream. Taking advantage of the reproducibility characteristic of good pseudorandom number generators, distributed processes can come to a common decision without global process synchronization. Similarly, the common decisions in which two processes will participate in replica exchange and whether the replica exchange attempt will be accepted can be made by using the same random number streams in multiple processes without communication among processes.

In our parallel implementation of the PT algorithm, multiple random number streams are used to minimize interprocessor communication; however, the problem of possible correlation among the random number streams arises. Intra-stream correlation will form sophisticated pattern, which may lead to defective or even erroneous results in Monte Carlo simulations. To avoid the intra-stream correlation problem, we employ the SPRNG library, which can produce up to 2^{78000} - 1 independent random number streams with sufficiently long period and good quality via appropriate parameterization. Properly configuring the random number generators in the SPRNG library, independence of the parallel random number streams used in a parallel PT implementation can be ensured [8], [12].

3.2 Configuration Exchange or Temperature Exchange in Replica Exchange?

Replica exchange is employed in the PT scheme for improving mixing among the Markov chains running at various temperature levels. Replica exchange requires

system passing configuration information between two processes carrying out the corresponding Markov chains. In many practical simulation applications, e.g., a large protein with hundreds of residues, or a physical system with thousands of molecules, replica exchange by swapping the system configurations will be rather costly because of large amount of interprocessor communication required. An alternative way to reduce the communication is to use temperature exchange instead. Compared to configuration exchange, temperature exchange only requires swapping of the temperature T_i, energy function value $E(x_i)$, and proposal distribution function $q_i(.|.)$ for index i, if different proposal functions are used in different processes. Temperature exchange only requires swapping of at most two floating point numbers and one integer index. As a result, temperature exchange is much more communication friendly than configuration exchange in complex system simulations.

3.3 Neighboring Replica Exchange or Random Replica Exchange?

If temperature exchange is used instead of configuration exchange for our replica exchange, the amount of interprocessor communication can be significantly reduced in complex systems with large amounts of configuration information. However, the temperature order is disturbed in temperature exchange, which is no longer ordered by process rank. As a result, after several steps of temperature exchange, swapping of neighboring processes does not lead to exchange of neighboring temperature levels. Performing replica exchange at neighboring temperatures requires global awareness of the temperature distribution at different processes, which demands additional global process synchronization by gathering the temperature values distributed on different processes.

Instead of replica exchange at neighboring temperature levels, a more general form of replica exchange is random replica exchange, where replica exchange takes place between any two randomly selected temperature levels, i and j.

$$x_i \leftrightarrow x_j . \tag{7}$$

Neighboring replica exchange is a special case of random replica exchange where $i = j + 1$. Accordingly, the exchange is accepted according to the Metropolis-Hastings criterion with probability

$$\begin{aligned} &P_{\text{Re}\,plica}(x_i \leftrightarrow x_j) = P(\{x_1,...,x_i,...,x_j,...,x_N\} \mid \{x_1,...,x_j,...,x_i,...,x_N\}) \\ &= \min(1, \frac{\Pi(\{x_1,...,x_j,...,x_i,...,x_N\})}{\Pi(\{x_1,...,x_i,...,x_j,...,x_N\})}) \\ &= \min(1, e^{-\beta_i E(x_j) - \beta_j E(x_i) + \beta_j E(x_j) + \beta_i E(x_i)}) \end{aligned} \tag{8}$$

Notice that the detailed balance condition still holds for random replica exchange transitions.

$$\begin{aligned} &P(\{x_1,...,x_i,...,x_j,...,x_N\} \mid \{x_1,...,x_j,...,x_i,...,x_N\})\Pi(\{x_1,...,x_j,...,x_i,...,x_N\}) \\ &= P(\{x_1,...,x_j,...,x_i,...,x_N\} \mid \{x_1,...,x_i,...,x_j,...,x_N\})\Pi(\{x_1,...,x_i,...,x_j,...,x_N\}) \end{aligned} \tag{9}$$

Two unbiased participant processes in random replica exchange can be determined by a shared global random number, where global synchronization is not necessary. The random replica exchange can be thought of as a "larger" replica transition step in PT, which allows replica exchange attempts at a larger temperature difference. However, random replica exchange will have a lower success rate compared to neighboring replica exchange. Yet, using large transition steps in combination with small transition steps usually results in reduced waiting time when a system is trapped by deep local minima in a MCMC evolution [10, 11].

3.4 Random Number Streams

Various independent SPRNG random number streams, including local streams and global streams, are involved in decision making in our parallel implementation of PT. These random number streams are shown in Table 1.

Table 1. Independent Random Number Streams and Their Roles in Decentralized PT Scheme

stream name	sharing	number	decision
proposal stream	local	*N*	Proposal new configuration xi' for local Metropolis transition
local acceptance stream	local	*N*	Acceptance of local transition according to Metropolis ratio
replica exchange stream	same in all processes	1	Whether to perform replica exchange at current time step
participant stream	same in all processes	1	Whether the current process should participate in replica exchange at this time
swap stream	same in any process pair	*N*(N*-1)/2	Acceptance of replica exchange transition according to exchange ratio

3.5 Efficient Parallel PT Implementation

Fig. 1 shows the flowchart of our decentralized replica exchange PT scheme. At the beginning, the system configuration, temperature, SPRNG random number generators, and other necessary variables are initialized in each process. In Metropolis transitions, random numbers from the proposal stream are used to produce a proposal transition and then a random number from the local acceptance stream is used to determine whether the proposal transition will be accepted. Both proposal stream and local acceptance stream are local streams which are different and independent in different processes. After a Metropolis transition, in each process, a random number from the replica exchange stream is drawn to decide whether a replica transition will be performed. Both the replica exchange stream and participant stream are globally shared, where random number sequences are exactly the same in all processes. If yes, random numbers are generated in the participant stream to determine which two processes will participate in replica exchange. The non-selected processes skip replica transition. For the two randomly selected participant processes, temperature and

energy function values are exchanged via the MPI function call MPI_Sendrecv(). A random number from the swap stream which is identical in both participant processes is drawn to decide whether the replica exchange attempt will be accepted. In this parallel PT implementation, the only interprocessor communication required is that for the exchange of temperature and the energy function value.

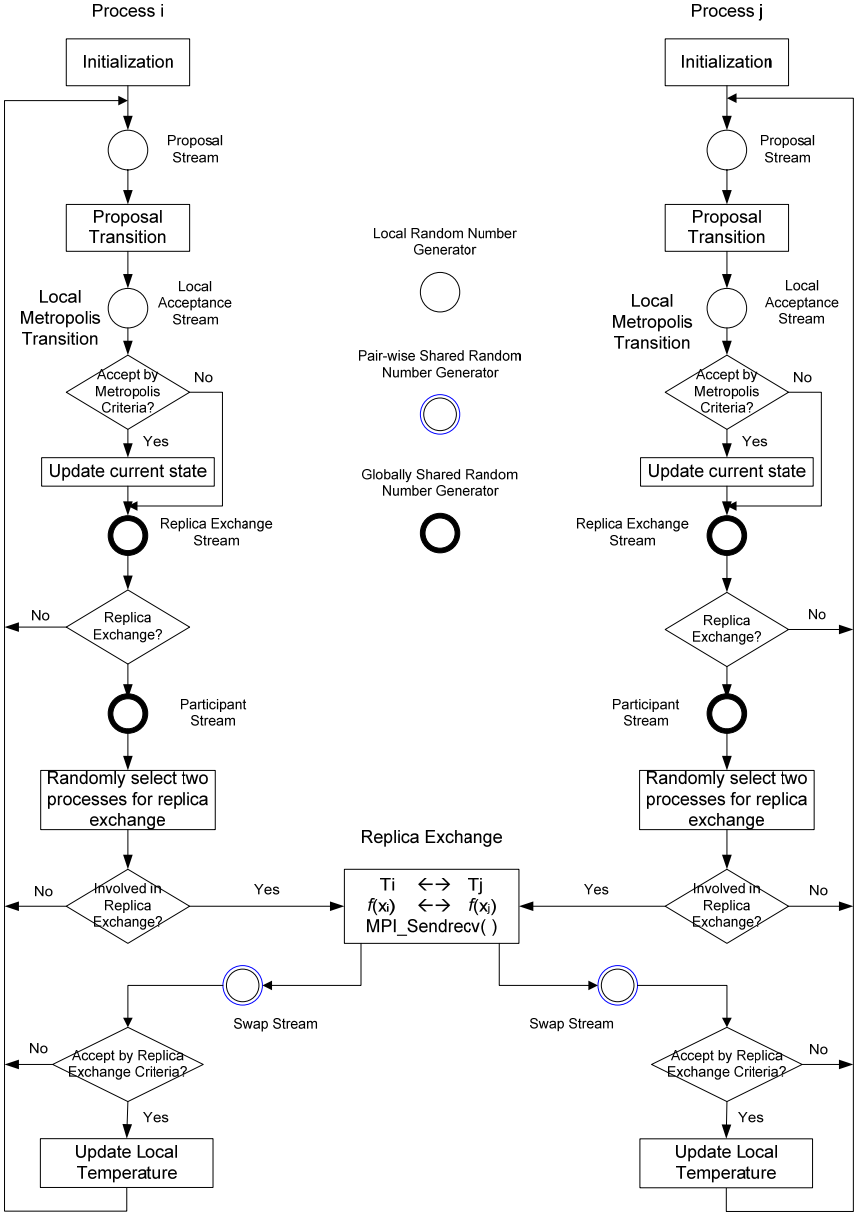


Fig. 1. Flowchart of the Decentralized Replica Exchange PT Scheme

3.6 Implementation Analysis

1. Efficiency
This parallel implementation of PT eliminates global synchronization operations in PT processes. Replica exchanges, provided that they involve different processes, can be executed in parallel. The amount of communication information is also minimized by using temperature exchange and randomly choosing participant processes.

2. Reproducibility
Notice that the simulation this parallel implementation of PT is reproducible. First of all, all SPRNG random number streams involved can be exactly reproduced by retrieving the same parameters and seeds in each pseudorandom number generator. Secondly, in each process, the Metropolis transition can be reproduced by reproducing the local random numbers in the proposal stream and the local acceptance stream. Thirdly, deciding when to perform replica exchange and the participant processes are reproducible by retrieving the global random number sequences of the replica exchange stream and participant stream, respectively. Finally, when a replica exchange is attempted, each process pair can be reproduced by reproducing the corresponding swap stream.

4 Computational Results

4.1 Rosenbrock's Function

The generalized n-dimensional Rosenbrock's function is defined as

$$f(x_1,...,x_n) = \sum_{i=2}^{n} (100(x_i - x_{i-1}^{\ 2})^2 + (1 - x_{i-1})^2) \quad , \tag{10}$$

where the global minimum is at (1.0, 1.0, …, 1.0). The Rosenbrock's function is a notorious benchmark function in optimization because of its slow convergence for

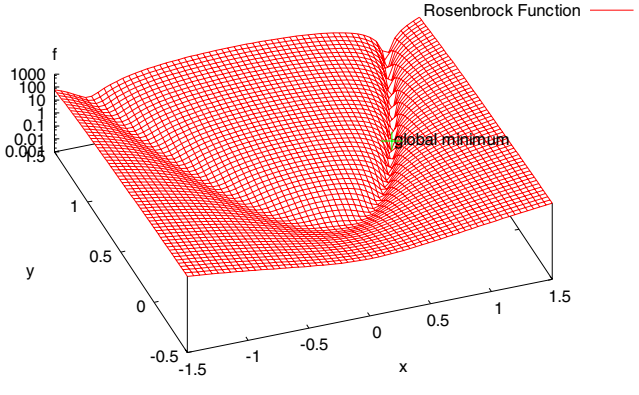


Fig. 2. Two-dimensional Rosenbrock's Function. Note the logarithmic scale of the function's axis.

most optimization methods. Due to a long narrow valley present in this function, gradient-based methods may have to spend a large number of iterations before the global minimum is reached. Fig. 2 shows the two dimensional Rosenbrock's function.

Fig. 3 shows the performance comparison of PT with decentralized replica exchange, PT using a master-slave paradigm [13], and parallel Metropolis in a 100-dimensional Rosenbrock's Function1. The replica exchange probability θ is 10%. In the master-slave parallel PT, each process carries out local Metropolis transitions and one process is designated as the master, which collects replica information from each slave process, performs replica exchange, and then scatters replica information back to each slave process. Global synchronizations are required in the master-slave paradigm. Parallel Metropolis is an naturally parallel implementation, where each process carries out a Metropolis transition, and no replica exchange takes place among the processes. From Fig. 3, one can see that the global synchronization operations in master-slave PT are costly and post heavy interprocessor communication overhead (358.3%). In contrast, the interprocessor communication overhead introduced by PT with decentralized replica exchange is small (approximately 10.2%) compared to that of the naturally parallel Metropolis in the Rosenbrock's function experiment. The resulting curves of the best, worst, and average objective function values over the number of iterations in 10 independent parallel Metropolis and PT runs are shown in Fig. 4. In this experiment, parallel Metropolis has the same initial position, temperature, and transition step size configuration as PT but does not carry out replication exchange between temperature levels. One can observe that PT exhibits a faster convergence to global minimum compared to parallel Metropolis due to reduction of relaxation time by the replication exchange transitions.

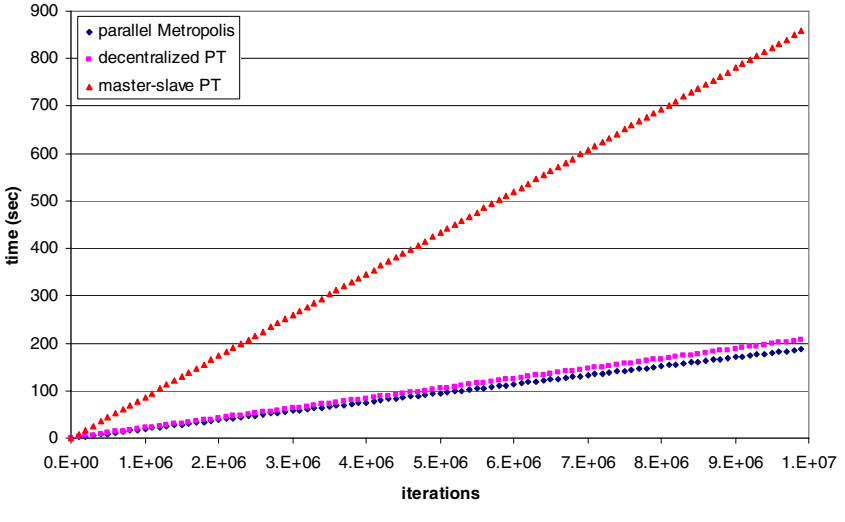


Fig. 3. Performance Comparison of Decentralized Parallel PT, Parallel Metropolis, and Master-Slave Parallel PT in Rosenbrock's Function Optimization on 8 processors

[1] The computations are carried out on a Beowulf Linux cluster with 8 2.2GHZ Xeon processors, with 1G Memory each node.

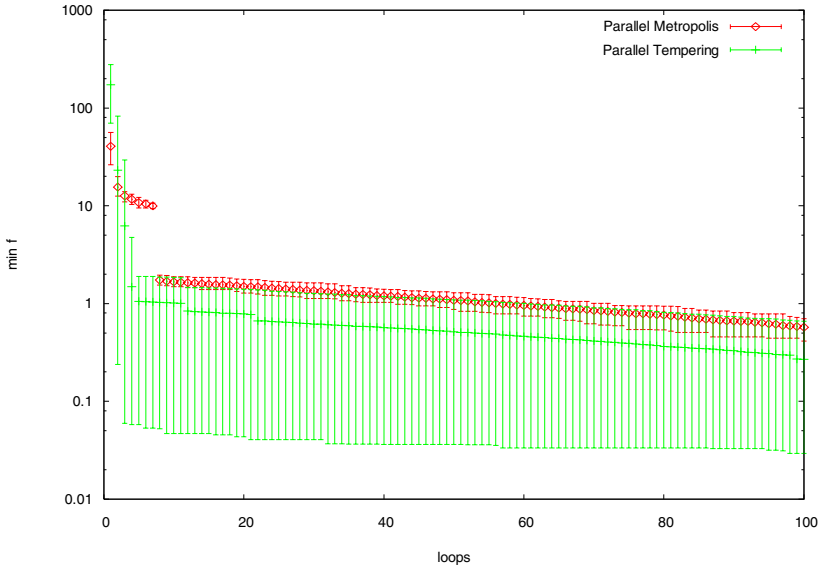


Fig. 4. Experimental Runs of Parallel Metropolis and Parallel Tempering on the 100-Dimensitonal Rosenbrock's Function. Each loop includes 10^5 iterations and shows the best, worst, and average function values.

4.2 Rugged Funnel-Like Function

In this experiment, we construct a "rugged" funnel-like function

$$E(x_1,...,x_n) = -\frac{c}{2n}\sum_{k=1}^{n}\left\{1 + A\sum_{i=0}^{m} a^i \cos(b^i x_k)\right\}, \tag{11}$$

where $A = (a\text{-}1)/(a^{m+1}\text{-}1)$, n is the dimension, and m, a, b, and c are some tunable constants to determine the depth of the funnel and the number of local minima along it.

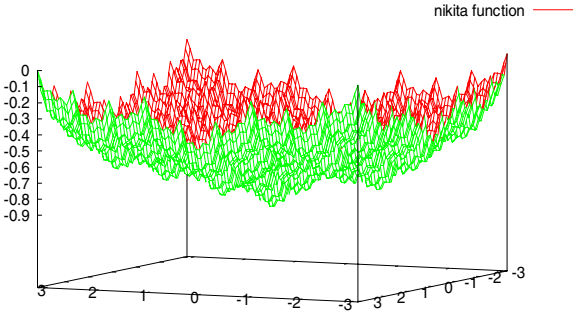


Fig. 5. A two-dimensional "Rugged" Funnel-like Function ($m = 4$, $a = 0.7$, $b = 3.0$, $c = 1.0$)

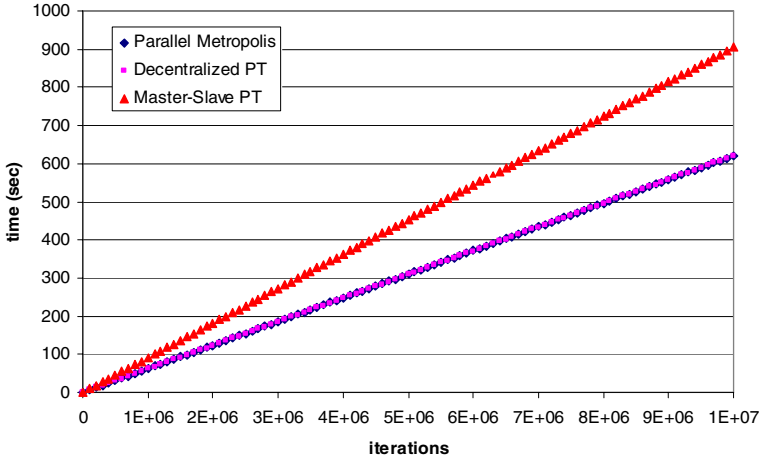


Fig. 6. Figure 6: Performance Comparison of Decentralized Parallel PT, Parallel Metropolis, and Master-Slave Parallel PT in Rugged Funnel-like Function Optimization on 8 processors

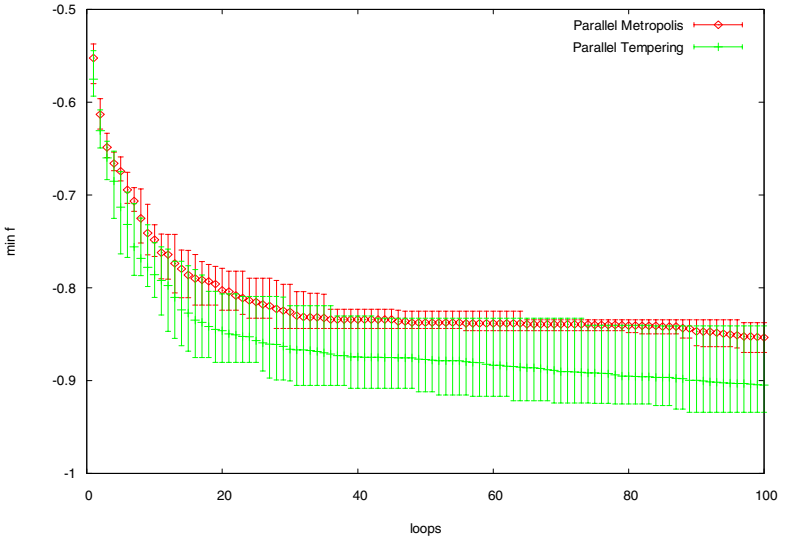


Fig. 7. Experimental Runs of Parallel Metropolis and Parallel Tempering on a 100-Dimensitonal Rugged Funnel-like Function. Each loop includes 105 iterations and shows the best, worst, and average function values.

The only global minimum is located at (0, …, 0) and is equal to -*c*. Fig. 5 shows the two-dimensional "rugged" funnel-like function where $c = 1.0$.

Fig. 6 shows the performance comparison of master-slave PT, PT with decentralized replica exchange and parallel Metropolis in a 100-dimensional rugged funnel-like function. Similar to the computational experiments on the Rosenbrock's function,

decentralized replica exchange PT yields almost indistinguishable interprocessor communication overhead (0.35%) compared to parallel Metropolis and outperforms master-slave PT. As the curves of the best, worst, and average objective function values over the number of iterations in 10 independent runs shown in Fig. 7, PT exhibits a faster convergence to global minimum (0.0) than parallel Metropolis.

Notice that PT is an effective sampling method, which may locate the valley leading to the global minimum but may not be able to actually approach global minimum in high precision. Extended MCMC algorithm, such as the hybrid PT/SA algorithm [14] or combining MCMC with local minimization (downhill) methods [15], can more aggressively minimize the solution.

5 Summary

In this article, we developed a decentralized PT implementation, using the MPI and SPRNG libraries. Taking advantage of the determinism and reproducibility characteristics of parallel pseudorandom number streams in SPRNG, and using temperature exchange instead of replica exchange, we are able to eliminate the need for global synchronization and to minimize interprocessor communication. Our computational experiments, based on applying the decentralized PT implementation to the high-dimensional Rosenbrock's function and rugged funnel-like function show that insignificant amount of interprocessor communication overhead contributed to the overall simulation time. Since this decentralized PT implementation can also be applied to some extended PT algorithms, such as hybrid Parallel Tempering/Simulated Tempering [14], adaptive PT [4], parallel sintering [17], and various Evolutionary Markov Chain Monte Carlo methods [16], [18], [19], [20], this seems like a likely avenue for future work.

Acknowledgments. Yaohang Li acknowledges support by the Ralph E. Powe Junior Faculty Enhancement Award of Oak Ridge Associated Universities. The research was also sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory (ORNL), managed by UT-Battelle, LLC for the U. S. Department of Energy under Contract No. DE-AC05-00OR22725

References

1. Marinari, E., Parisi, G.: Simulated Tempering: a New Monte Carlo Scheme. Europhysics Letters 19, 451–458 (1992)
2. Geyer, C.J., Thompson, E.A.: Annealing Markov Chain Monte Carlo with Applications to Ancestral Inference. Journal of the American Statistical Association 90, 909–920 (1995)
3. Falcioni, M., Deem, M.W.: A Biased Monte Carlo Scheme for Zeolite Structure Solution. J. Chem. Phys 110, 1754–1766 (1999)
4. Schug, A., Herges, T., Verma, A., Wenzel, W.: Investigation of the parallel tempering method for protein folding. J. Phys: Condens. Matter 17, 1641–1650 (2005)
5. Sikorski, A.: Properties of Star-Branched Polymer Chains – Application of the Replica Exchange Monte Carlo Method. Macromolecules 35(18), 7132–7137 (2002)

6. Schug, A., Wenzel, W.: Predictive in-silico all atom folding of a four helix protein with a free energy model. J. Am. Chem. Soc. 126, 16737 (2004)
7. Li, Y., Strauss, C.E.M., Gorin, A.: Parallel Tempering in Rosetta Practice. In: Zhang, D., Jain, A.K. (eds.) ICBA 2004. LNCS, vol. 3072, Springer, Heidelberg (2004)
8. Mascagni, M., Srinivasan, A.: Algorithm 806: SPRNG: A Scalable Library for Pseudorandom Number Generation. ACM Transactions on Mathematical Software 26, 436–461 (2000)
9. Bu, L., Straub, J.E.: Simulating Vibrational Energy Flow in Proteins: Relaxation Rate and Mechanism for Heme Cooling in Cytochrome c. J. Phys. Chem. B 107, 12339–12345 (2003)
10. Li, Y., Protopopescu, V.A., Gorin, A.: Accelerated Simulated Tempering. Physics Letters A 328(4), 274–283 (2004)
11. Liu, J.S., Liang, F., Wong, W.H.: The Use of Multiple-Try Method and Local Optimization in Metropolis Sampling. Technical Report, Department of Statistics, Stanford University (1998)
12. Srinivasan, A., Mascagni, M., Ceperley, D.: Testing Parallel Random Number Generators. Parallel Computing 29, 69–94 (2003)
13. Li, Y., Clark, J., Zhang, X.: Parallel Implementation of the Accelerated Simulated Tempering Method. Proceedings of 3rd NPSC Conference, Atlanta (2006)
14. Li, Y., Protopopescu, V.A., Arnold, N., Zhang, X., Gorin, A.: Hybrid Parallel Tempering/Simulated Annealing Method. submitted to Physical Review E (2006)
15. Du, Z., Li, S., Li, S., Wu, M., Zhu, J.: Massively parallel simulated annealing embedded with downhill – a SPMD algorithm for cluster computing. In: Proceedings of 1st IEEE Computer Society International Workshop on Cluster Computing (1999)
16. Drugan, M.M., Thierens, D.: Evolutionary Markov Chain Monte Carlo. Technical Report UU-CS-2003-047, Utrecht university (2003)
17. Liu, J.S., Sabatti, C.: Simulated Sintering: Markov Chain Monte Carlo with Spaces of Varying Dimensions. In: Bayesian Statistics 6, pp. 389–413. Oxford University Press, Oxford (1999)
18. Cercueil, A., Francois, O.: Monte Carlo simulation and population-based optimization. In: Congress on Evolutionary Computation, pp. 191–198 (2001)
19. Laskey, K.B., Myers, J.W.: Population Markov Chain Monte Carlo. Machine Learning , 175–196 (2003)
20. Mahfoud, S.W., Goldberg, D.E.: Parallel Recombinative Simulated Annealing: a Genetic Algorithm. Parallel Computing, 1–28 (1995)

Approximation Algorithms for 2-Source Minimum Routing Cost k-Tree Problems

Yen Hung Chen[1,⋆], Gwo-Liang Liao[1], and Chuan Yi Tang[2]

[1] Department of Information Science and Management Systems, National Taitung University, 684, Sec.1, Chunghua Rd., Taitung 950, Taiwan, R.O.C.
yhchen@nttu.edu.tw, glliao@nttu.edu.tw

[2] Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan, R.O.C.
cytang@cs.nthu.edu.tw

Abstract. In this paper, we investigate some k-tree problems of graphs with given two sources. Let $G = (V, E, w)$ be an undirected graph with nonnegative edge lengths and two sources $s_1, s_2 \in V$. The first problem is the 2-*source minimum routing cost k-tree* (2-kMRCT) problem, in which we want to find a tree $T = (V_T, E_T)$ spanning k vertices such that the total distance from all vertex in V_T to the two sources is minimized, i.e., we want to minimize $\sum_{v \in V_T}\{d_T(s_1, v) + d_T(s_2, v)\}$, in which $d_T(s, v)$ is the length of the path between s and v on T. The second problem is the 2-*source bottleneck source routing cost k-tree* (2-kBSRT) problem, in which the objective function is the maximum total distance from any source to all vertices in V_T, i.e., $\max_{s \in (s_1, s_2)}\{\sum_{v \in V_T} d_T(s, v)\}$. The third problem is the 2-*source bottleneck vertex routing cost k-tree* (2-kBVRT) problem, in which the objective function is the maximum total distance from any vertex in V_T to the two sources , i.e., $\max_{v \in V_T}\{d_T(s_1, v) + d_T(s_2, v)\}$. In this paper, we present polynomial time approximation schemes (PTASs) for the 2-kMRCT and 2-kBVRT problems. For the 2-kBSRT problem, we give a $(2 + \varepsilon)$-approximation algorithm for any $\varepsilon > 0$.

Keywords: combinatorial optimization problem, k-tree, approximation algorithm, polynomial time approximation scheme (PTAS).

1 Introduction

Finding communication cost trees of a given graph is an important problem in network design. The communication cost is sum over the distance of all pairs of vertices u and v times their requirements between u and v. Given a graph, the *minimum communication cost spanning tree* (MCT) problem [12] is to find a spanning tree such that the communication cost is minimized and the *minimum routing cost spanning tree* (MRCT) problem [9,16] is a special case of the MCT problem, in which the requirement of each pair of vertices is one.

⋆ The Corresponding author.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 520–533, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Give a graph G with n vertices and a specified number k, the k-tree is the tree of G that spans at least k vertices, $k \leq n$. R. Ravi et al. [14] posted two problems, called as the *k-minimum spanning tree (kMST, also called k-tree)* and the *minimum communication cost k-tree* (kMCT) problems, respectively. The kMST (respectively, kMCT) problem is to find a k-tree such that the total edge length (respectively, communication cost) is minimized. For the kMST problem, they showed this problem is NP-hard and gave a $2\sqrt{k}$-approximation algorithm. For the kMCT problem, they also showed this problem is NP-hard and can't be approximated to within any factor unless $P = NP$. It is not difficult that an optimal solution of the two problems will be a tree spanning exactly k vertices. Hence, a k-tree is defined that a tree spans exactly k vertices. In the past many papers focused the kMST problem and constant-approximation algorithms had been proposed in series works [1,2,3,10]. However, some communication cost problems are still open.

Motivated by applications of multicasting and broadcasting, several multi-source spanning tree problems have been studied [4,5,6,8,17]. In these problems, a network is modeled by an undirected graph with nonnegative edge length function. Some vertices are defined as destinations, whereas some of them are sources. The goal is to find a tree connecting all sources and destinations such that packets can be routed from sources to destinations along the tree edges. The length of an edge usually signifies the cost of routing a packet along the edge, and the cost from a source to a destination is the distance between the two vertices, i.e., the sum of edge lengths of the unique path between the two vertices in the tree. With different applications, the objective functions of the problems are defined differently but in general to minimize some distance related costs.

Given an undirected graph $G = (V, E, w)$ with nonnegative edge length function w and $S \subset V$ a set of $\Re$ specified sources, the $\Re$-source MRCT ($\Re$-MRCT) problem is to find a spanning tree T minimizing $\sum_{s \in S}\{\sum_{v \in V} d_T(s, v)\}$, where $d_T(s, v)$ is the length of the path between s and v in T. If there is only one source, the problem is called the *shortest paths tree* problem, i.e., a spanning tree in which the path from the source to each vertex is a shortest path on the given graph. This problem has been well studied and most efficient algorithms run in $O(|E| + |V| \log |V|)$ time [7].

In the case of $\Re > 1$, the $\Re$-MRCT problem is a generalization of the MRCT problem, in which all vertices are sources. The MRCT problem is NP-hard [9,13]. the $\Re$-MRCT problem for general $\Re$ is obviously NP-hard since the MRCT problem is a special case of the $\Re$-MRCT. The NP-hardness of the 2-MRCT problem was shown independently in [8] and [17] and a PTAS was also proposed [17]. The $\Re$-MRCT problem for any constant $\Re$ was also shown to be NP-hard [18] and a 2-approximation algorithm can be found [15].

While the $\Re$-MRCT problem is defined by a min-sum objective function, Connamacher and Proskurowski [6] posted two variants of the $\Re$-MRCT problem with bottleneck objective functions. They are the $\Re$-*source maximum vertex shortest paths spanning tree* ($\Re$-MVST) (also called the $\Re$-*source bottleneck vertex routing cost spanning tree* ($\Re$-BVRT) in [4]) problem and the $\Re$-*source maximum*

source shortest paths spanning tree (ℜ-MSST) (also called the ℜ-*source bottleneck source routing cost spanning tree* (ℜ-BSRT) in [4]) problem, in which the objective functions are $\max_{v\in V}\{\sum_{s\in S} d_T(s,v)\}$ and $\max_{s\in S}\{\sum_{v\in V} d_T(s,v)\}$, respectively. Both problems were shown to be NP-Complete in the strong sense even for $\Re = 2$ [6]. Chen, Wu, and Tang [4,5] showed a PTAS for the 2-BVRT, a $(2+\varepsilon)$-approximation algorithm for the 2-BSRT and 3-approximation algorithms for the ℜ-BVRT and ℜ-BSRT, when $\Re > 2$.

In this paper, we will focus the k-tree versions of the ℜ-MRCT, ℜ-BVRT and ℜ-BSRT problems, called as ℜ-*source minimum routing cost k-tree* (ℜ-kMRCT), ℜ-*source bottleneck vertex routing cost k-tree* (ℜ-kBVRT) and ℜ-*source bottleneck source routing cost k-tree* (ℜ-kBSRT) problems, respectively. It is clear that feasible solutions for these three problems must contain the ℜ sources.

That is not difficult to show these three problems are NP-hard even for $\Re = 2$. We focus on the approximation algorithms for these three problems. For the 2-kMRCT and 2-kBVRT problems, we will present PTASs. For the 2-kBSRT problem, we will show a $(2+\varepsilon)$-approximation algorithm, for $\varepsilon > 0$.

The rest of this paper is organized as follows. In Section 2, some definitions and notations are given. In Section 3, we describe a PTAS for the 2-kMRCT problem. In Section 4, we show a $(2+\varepsilon)$-approximation algorithm for the 2-kBSRT. In section 5, we also present a PTAS for the 2-kBVRT. Finally, we make a conclusion in Section 6.

2 Preliminaries

In this paper, a graph is simple, connected and undirected. By $G = (V, E, w)$, we denote a graph G with vertex set V, edge set E and edge length function w. The edge length function is assumed to be nonnegative. For any graph G, $V(G)$ denotes its vertex set and $E(G)$ denotes its edge set. For a subgraph $\mathcal{G}$ of G, we define $w(\mathcal{G}) = w(E(\mathcal{G})) = \sum_{e\in E(\mathcal{G})} w(e)$. We shall also use n to denote $|V(G)|$. Given an integer k, $k \le n$, a k-tree T of G is a tree spanning k vertices in the G.

Definition 1. *For $u, v \in V$, $SP_G(u,v)$ denotes a shortest path between u and v on G. The shortest path length is denoted by $d_G(u,v) = w(SP_G(u,v))$.*

Definition 2. *Let $\mathcal{G}$ be a subgraph of G. For a vertex $v \in V(G)$, we let $d_G(v,\mathcal{G})$ denote the shortest distance from v to $\mathcal{G}$, i.e., $d_G(v,\mathcal{G}) = \min_{u\in V(\mathcal{G})}\{d_G(v,u)\}$. The definition also includes the case that $\mathcal{G}$ is a subset of V.*

Definition 3. *Let $T = (V_T, E_T)$ be a tree of G. For $v \in V_T$ and a vertex set $\mathcal{L} \subset V_T$, we use $D_T(v,\mathcal{L})$ to denote the total distance from vertex v to $\mathcal{L}$, i.e., $D_T(v,\mathcal{L}) = \sum_{l\in\mathcal{L}} d_T(v,l)$.*

Definition 4. *Let T be a k-tree with $S \subset V(T)$ a set of ℜ sources. The* routing cost *of T with source set S, denoted by $C_R(T)$, is the total distances from all vertices of T to all sources, i.e., $\sum_{v\in V(T)} D_T(v,S)$. Given a graph and a set of ℜ sources, the* minimum routing cost k-tree *(ℜ-kMRCT) problem is to find a k-tree T spanning exactly ℜ sources and $k - \Re$ vertices in $V \setminus S$ with minimum $C_R(T)$ among all possible trees.*

Definition 5. *Let T be a k-tree with $S \subset V(T)$ a set of $\Re$ sources. The* bottleneck source routing cost *of T, denoted by $C_S(T)$, is the maximum total distances from any source to all vertices of T, i.e.,* $\max_{s \in S} D_T(s, V(T))$. *Given a graph and a set of $\Re$ sources, the* bottleneck source routing cost k-tree *($\Re$-kBSRT) problem is to find a k-tree T spanning exactly $\Re$ sources and $k - \Re$ vertices in $V \setminus S$ with minimum $C_S(T)$ among all possible trees.*

Definition 6. *Let T be a k-tree with $S \subset V(T)$ a set of $\Re$ sources. The* bottleneck vertex routing cost *of T, denoted by $C_V(T)$, is the maximum total distances from any vertex of T to all sources, i.e.,* $\max_{v \in V(T)} D_T(v, S)$. *Given a graph and a set of $\Re$ sources, the* bottleneck vertex routing cost k-tree *($\Re$-kBVRT) problem is to find a k-tree T spanning exactly $\Re$ sources and $k - \Re$ vertices in $V \setminus S$ with minimum $C_V(T)$ among all possible trees.*

Let G be a graph and $r \in V(G)$. A *shortest path tree* rooted at r is a spanning tree T of G such that $d_T(r, v) = d_G(r, v)$ for each vertex $v \in V$. That is, on a shortest path tree, the path from the root to any vertex is a shortest path in G. Let $\mathcal{L} \subset V$. A forest F is a *shortest path forest* with multiple roots $\mathcal{L}$ if $d_F(v, \mathcal{L}) = d_G(v, \mathcal{L})$ for any vertex $v \in V$, i.e., each vertex is connected to the closest root by a shortest path. A shortest path forest [17] can be constructed by an algorithm similar to the shortest path tree algorithm [7]. The most efficient algorithm of this problem runs in $O(|E| + |V| \log |V|)$ time.

Given a graph $G = (V, E, w)$ with a source vertex s, a destination vertex t and an integer h which is number of hops (vertices), the *h-hop (vertex) constrained shortest path* problem is to find a path P in G from s to t with the minimum total length such that the number of vertices in P are at most $h + 1$. Note that if $h_1 < h_2$, the total length of h_2-hop constrained shortest path is small than or equal to the total length of h_1-hop constrained shortest path. Given an integer H, $H < n$, a source s and a destination t, the *all hops shortest path* problem is to find, for each hop value h, $1 \leq h \leq H$, a h-hop constrained shortest path between s and t if such a path does exist. Guérin and Orda [11] showed this problem can be solved by *Bellman-Ford shortest path* algorithm [7], in which the time complexity is $O(H|E|)$ time. Note that given one source s, the i-iteration of *Bellman-Ford* algorithm identifies an optimal i-hop constraint shortest path between s and each destination t in V. It runs H-iteration for the *all hops shortest path* problem.

For this paper, we assume that s_1 and s_2 are the two given sources. The next lemma comes from [17].

Lemma 1. *Suppose that $P = \{s_1 = p_1, p_2, \ldots, s_2 = p_\lambda\}$ is a tree T of G between s_1 and s_2. Let m_1, m_2,... ,m_i be i vertices such that P connects the consecutive m_j. Given two sources s_1, s_2 and the i-tuple $(m_1, m_2, \ldots, m_i)$, there is an $O(in^2)$ time algorithm which constructs a tree X spanning the given vertices and having the property that $d_X(v, s_1) + d_X(v, s_2) \leq w(P)$ for any $v \in V(X)$.*

However, this lemma can't work on k-tree problems, since the tree X may has more than k vertices. Hence, we modify above lemma such that this tree X has at most k vertices.

Lemma 2. *Let $P = \{s_1 = p_1, p_2, \ldots, s_2 = p_\lambda\}$ be a path of a k-tree T and m_1, m_2,... ,m_i be i vertices such that P connects the consecutive m_j. Given the two sources s_1, s_2 and the i-tuple $(m_1, m_2, \ldots, m_i)$, there is an $O(ik^{i+1}n^2)$ time algorithm which constructs a tree X spanning the given vertices such that $|V(X)|$ at most k vertices and having the property that $d_X(v, s_1) + d_X(v, s_2) \leq w(P)$ for any $v \in V(X)$.*

Proof. Let $m_0 = s_1$ and $m_{i+1} = s_2$. For any $0 \leq j \leq i$, let $M_{j,h}$ denote a h-hop constrained shortest path between m_j and m_{j+1}. Let $H = k - 1$, we first find the *all hops shortest path* problem between m_j and m_{j+1} for $0 \leq j \leq i$. Let a path collection $PC = \{M_{0,\hbar_0}, M_{1,\hbar_1}, \ldots, M_{i,\hbar_i}\}$ denote a path set such that $\hbar_0 + \hbar_1 + \ldots + \hbar_i \leq k - 1$. Then, for each path collection PC, we find a tree X and $V(X) \subseteq V(PC)$ such that $d_X(v, s_1) + d_X(v, s_2) \leq w(P)$ for any $v \in V(X)$. It can be designed by the previous algorithm [17]. We state the result but the proof is omitted.

Now, we analysis the time complexity of this algorithm. First, finding the *all hops shortest path* between m_j and m_{j+1} takes $O(k|E|)$ time. Hence, totally we need $O((i+1)k|E|)$ time. Then we have at most $O(k^{i+1})$ path collection and for each path collection, it takes $O(in^2)$ time to find the tree X. Hence, the time complexity of this algorithm is $O(ik^{i+1}n^2)$. □

3 A PTAS for the 2-kMRCT Problem

In this section, we show a PTAS for the 2-kMRCT problem. The PTAS is a modification of the one designed in [17] for the 2-MRCT. First, we show that the simple 2-approximation algorithm for the 2-kMRCT problem. We first finds a shortest path P_T from s_1 to s_2 subject to $|V(P_T)| \leq k$. If no such path exists, there is no solution. Then, finding a shortest path forest F of G with multiple root P_T. Let a spanning tree $T' = F \cup P_T$. Moreover, we partition $V \setminus V(P_T)$ into two vertices sets $\mathcal{L}_1$ and $\mathcal{L}_2$ such that $|\mathcal{L}_1| = k - |V(P_T)|$, $|\mathcal{L}_2| = n - k$ and $d_G(l_1, P_T) \leq d_G(l_2, P_T)$ for any $l_1 \in \mathcal{L}_1$ and $l_2 \in \mathcal{L}_2$. We output the k-tree T which is the subtree of T' induced by $\mathcal{L}_1 \cup V(P_T)$. The following property is useful for showing the performance of the algorithm.

Lemma 3. *Let T be any k-tree of $G = (V, E, w)$ and P be the path between s_1 and s_2 on T. $C_R(T) = k * w(P) + 2\Sigma_{v \in V(T)}\{d_T(v, P)\}$.*

Proof. First, by definition, $D_T(v, S) = d_T(v, s_1) + d_T(v, s_2) = w(P) + 2d_T(v, P)$ for any $v \in V(T)$. The result can be showed by summer over all vertices in $V(T)$. □

Now we show the performance of the simple algorithm.

Lemma 4. *A 2-approximation algorithm for the 2-kMRCT can be found in $O(k|E| + n \log n)$ time.*

Proof. First we show a lower bound of the optimum. Let Y and T be the optimal and approximation trees of the 2-kMRCT problem. For any vertex $v \in V(Y)$,

$$C_R(Y) = \sum_{v \in V(Y)} D_Y(v, S) \geq 2 \sum_{v \in V(Y)} d_Y(v, S) \geq 2 \sum_{v \in V(Y)} d_G(v, S). \quad (1)$$

Let P_Y be the path between s_1 and s_2 on Y. Since P_T is the shortest path between the two sources subject to $|P_T| \leq k$, hence $w(P_T) \leq w(P_Y)$. By Lemma 3, we have

$$C_R(Y) \geq k * w(P_Y) \geq k * w(P_T). \quad (2)$$

By equations (1) and (2), we have $C_R(Y) \geq \frac{1}{2}(kw(P_Y) + 2\sum_{v \in V(Y)} d_G(v, S))$. For any $v \in \mathcal{L}_1$, $D_T(v, S) = 2d_T(v, P_T) + w(P_T)$. Hence, we have $C_R(T) = 2\sum_{v \in \mathcal{L}_1} d_G(v, P_T) + k * w(P_T)$. Since

$$\begin{aligned} \sum_{v \in \mathcal{L}_1} d_G(v, P_T) &= \sum_{v \in \{\mathcal{L}_1 \cap V(Y)\}} d_G(v, P_T) + \sum_{v \in \{\mathcal{L}_1 \setminus V(Y)\}} d_G(v, P_T) \\ &\leq \sum_{v \in \{\mathcal{L}_1 \cap V(Y)\}} d_G(v, P_T) + \sum_{v \in \{V(Y) \setminus V(T)\} \subseteq \mathcal{L}_2} d_G(v, P_T) \\ &\leq \sum_{v \in V(Y)} d_G(v, P_T), \end{aligned}$$

we have $C_R(T) = 2\sum_{v \in \mathcal{L}_1} d_G(v, P_T) + k * w(P_T) \leq 2\sum_{v \in V(Y)} d_G(v, P_T) + k * w(P_T) \leq 2\sum_{v \in V(Y)} d_G(v, S) + k * w(P_Y) \leq 2C_R(Y)$. Hence, T is a 2-approximation solution. Note that the time complexity of this algorithm is dominated by finding k-hop shortest path problem and one of constructing the shortest path forest. □

Now, we describe the PTAS and its performance analysis. Similar to the previous PTAS for the 2-MRCT, our PTAS tries to guess some vertices of the path between the two sources. For each guess, it first constructs a tree X spanning those guessed vertices and $|V(X)| \leq k$. Then, we extend the tree X to select $k - |V(X)|$ vertices by adding shortest paths to X. The performance is ensured by showing that at least one of the constructed trees is a good approximation solution and the approximation ratio approaches to 1 as the number of guessed vertices increasing. In the remaining paragraphs of this section, let Y be the optimal tree of the 2-kMRCT. Also let $P = (p_1 = s_1, p_2, p_3, \ldots, p_\lambda = s_2)$ be the path between the two sources on Y. Now we introduce a partition of the vertices, which appeared in the previous work for the 2-MRCT problem (Fig. 1).

Define V_i, $1 \leq i \leq \lambda$, as the set of the vertices connected to P at p_i and also let $p_i \in V_i$. Let $\rho \geq 0$ be an integer parameter to be determined later. For $0 \leq i \leq \rho + 1$, define $m_i = p_j$ in which j is the minimal index such that

$$\sum_{1 \leq q \leq j} |V_q| \geq \lceil i \frac{k}{\rho + 1} \rceil.$$

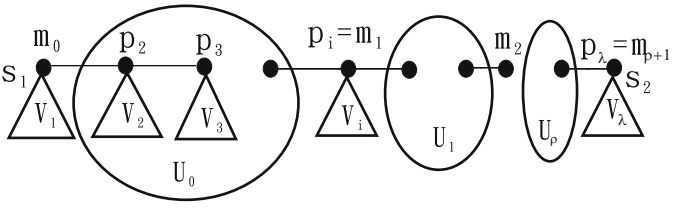


Fig. 1. The definitions of the partition of the vertices

By definition, $s_1 = m_0$ and $s_2 = m_{\rho+1}$. For $0 \le i \le \rho$, let $U_i = \bigcup_{a<j<b} V_j$ and P_i be the path from p_a and p_b, in which $p_a = m_i$ and $p_b = m_{i+1}$. Also let $U = V(Y) \setminus \bigcup_{0\le i\le\rho} U_i$ and $M = \{m_i : 0 \le i \le \rho + 1\}$. Note that the above definitions include the case $m_i = m_{i+1}$. In such a case, P_i contains only one vertex and U_i is empty. By the above definitions, the vertex set V is partitioned into $U, U_0, U_1, \ldots, U_\rho$ satisfying the following properties. The next lemma comes from [17]. We state the result but the proof is omitted.

Lemma 5. $\sum_{v\in U_i} d_G(v, M) \le \sum_{v\in U_i} d_Y(v, P) + \frac{k}{2(\rho+1)} w(P_i)$, *for any* $0 \le i \le \rho$ *and* $\sum_{v\in U} d_G(v, M) \le \sum_{v\in U} d_Y(v, P)$.

Our algorithm tries to correctly guess $m_1, m_2, \ldots, m_\rho$ and constructs a tree X spanning these vertices with the property that $d_X(v, s_1) + d_X(v, s_2) \le w(P)$ for any $v \in V(X)$ and $|V(X)| \le k$. Once such a tree X has been constructed, finding a shortest path forest F of G with multiple root $V(X)$. Let a spanning tree T' be the union of the forest F and X. For the tree T', we partition $V \setminus V(X)$ into two vertices sets $\mathcal{L}_1$ and $\mathcal{L}_2$ such that $|\mathcal{L}_1| = k - |V(X)|$, $|\mathcal{L}_2| = n - k$ and $d_G(l_1, X) \le d_G(l_2, X)$ for any $l_1 \in \mathcal{L}_1$ and $l_2 \in \mathcal{L}_2$. Let the k-tree T be the subtree of T' induced by $\mathcal{L}_1 \cup V(X)$. In the next lemma, we show that T is a $(\frac{\rho+2}{\rho+1})$-approximation of the 2-kMRCT.

Lemma 6. *Let X be a tree spanning M and $d_X(v, s_1) + d_X(v, s_2) \le w(P)$ for any $v \in V(X)$ with $|V(X)| \le k$. The k-tree T, which is a subtree of the spanning tree T' induced by $\mathcal{L}_1 \cup V(X)$, is a $(\frac{\rho+2}{\rho+1})$-approximation solution of the 2-kMRCT problem.*

Proof. By definition, the vertices $m_1, m_2, \ldots, m_\rho$ partition the vertex set $V(Y)$ into $(U, U_0, U_1, \ldots, U_\rho)$ and the subsets satisfy the properties in Lemma 5. Since $d_X(v, s_1) + d_X(v, s_2) \le w(P)$ for any $v \in V(X)$, we have

$$\begin{aligned} C_R(T) &= \left\{ \sum_{v\in V(T)} d_T(v, s_1) + d_T(v, s_2) \right\} \le 2 \sum_{v\in V(T)} d_T(v, X) + k\ w(P) \\ &= 2 \sum_{v\in V(X)} d_T(v, X) + 2 \sum_{v\in\mathcal{L}_1} d_T(v, X) + k\ w(P) \\ &= 2 \sum_{v\in\{\mathcal{L}_1\cap V(Y)\}} d_T(v, X) + 2 \sum_{v\in\{\mathcal{L}_1\setminus V(Y)\}} d_T(v, X) + k\ w(P). \qquad (3) \end{aligned}$$

For each $v \in \{\mathcal{L}_1 \setminus V(Y)\}$, there exists a unique vertex $y \in \{V(Y) \setminus V(T)\}$ such that $d_G(v, X) \leq d_G(y, X)$, since $y \in \mathcal{L}_2$. The equation (3) will be

$$C_R(T) \leq \sum_{v \in V(Y)} 2d_G(v, X) + k\ w(P). \tag{4}$$

For $v \in U$, since $d_G(v, X) \leq d_G(v, M)$ and by Lemma 5, we have

$$\sum_{v \in U} d_G(v, X) \leq \sum_{v \in U} d_G(v, M) \leq \sum_{v \in U} d_Y(v, P). \tag{5}$$

Similarly $d_G(v, X) \leq d_G(v, M)$ for $v \in U_i$, and then by Lemma 5, we have

$$\begin{aligned} \sum_{0 \leq i \leq \rho} \sum_{v \in U_i} d_G(v, X) &\leq \sum_{0 \leq i \leq \rho} \left(\sum_{v \in U_i} d_Y(v, P) + \frac{k}{2(\rho+1)} w(P_i) \right) \\ &\leq \sum_{0 \leq i \leq \rho} \sum_{v \in U_i} d_Y(v, P) + \frac{k}{2(\rho+1)} w(P). \end{aligned} \tag{6}$$

By equations (4)–(6) and Lemma 3,

$$\begin{aligned} C_R(T) &\leq \sum_{v \in V(Y)} 2d_G(v, X) + k\ w(P) \\ &\leq 2 \sum_{v \in U} d_Y(v, P) + 2 \sum_{0 \leq i \leq \rho} \sum_{v \in U_i} d_Y(v, P) + \frac{k}{(\rho+1)} w(P) + k\ w(P) \\ &= 2 \sum_{v \in V(Y)} d_Y(v, P) + \frac{\rho+2}{(\rho+1)} k\ w(P) \leq \left(\frac{\rho+2}{\rho+1} \right) C_R(Y), \end{aligned}$$

since $C_R(Y) = kw(P) + 2 \sum_{v \in V(Y)} d_Y(v, P)$. □

By Lemma 6, we can design a good approximation algorithm for the 2-kMRCT if the input graph is a metric graph. A metric graph is a complete graph with edge lengths satisfying the triangle inequality. In a metric graph, the edge between any pair of vertices is a shortest path. For a metric graph and an integer ρ, we try each possible path $(s_1, m_1, m_2, \ldots, m_i, s_2)$ for $i \leq \rho \leq k-2$. There exists at least one such paths whose internal vertices are also on the path P of the optimal tree in the same order. Since the graph is a metric graph, the length of the path is no more than $w(P)$. Connecting all other vertices to the path by a shortest path forest and select $(k - |V(P)|)$ vertices which have smaller shortest paths to M. We have a k-tree and it is a $(\rho+2)/(\rho+1)$-approximation solution of the 2-kMRCT. However, a problem is encountered when the input is a general graph instead of a metric graph. Hence, we need to apply Lemma 2. For a guessed i-tuple $(m_1, m_2, \ldots, m_i)$ of vertices, $i \leq \rho$, by Lemma 2, we can construct the tree X passing s_1, s_2 and the i-tuple such that $|V(X)| \leq k$ and $d_X(v, s_1) + d_X(v, s_2) \leq w(P)$ for any $v \in V(X)$.

We list the PTAS below.

Algorithm PTAS-2-kMRCT
Input: A graph $G = (V, E, w)$ and $s_1, s_2 \in V$, an integer $\rho \geq 0$, and $k > 0$.
Output: A k-tree T of G.
1. For each $i \leq \rho$ and i-tuple $(m_1, m_2, \ldots m_i)$ of vertices **do**
1.1 Let $m_0 = s_1$ and $m_{i+1} = s_2$.
1.2 For $0 \leq j \leq i$, let $H = k$ then find all hops shortest path from m_j to m_{j+1}.
1.3 Find a tree X as in Lemma 2.
1.4 Finding a shortest path forest F of G with multiple root $V(X)$. Let the spanning tree T' be the union of the forest F and X.
1.5 For the tree T', partition $V \setminus V(X)$ into two vertices sets $\mathcal{L}_1$ and $\mathcal{L}_2$ such that $|\mathcal{L}_1| = k - |V(X)|$, $|\mathcal{L}_2| = n - k$ and $d_G(l_1, X) \leq d_G(l_2, X)$ for any $l_1 \in \mathcal{L}_1$ and $l_2 \in \mathcal{L}_2$.
1.6 Let the k-tree T be the subtree of T' induced by $\mathcal{L}_1 \cup V(X)$.
end for
2. Output the k-tree T with minimum $C_R(T)$ among all constructed trees.

The result of this section is summarized in the following theorem.

Theorem 1. *The 2-kMRCT problem admits a PTAS. For any constant $\varepsilon > 0$, a (1+ε)- approximation algorithm of the 2-kMRCT of a graph G can be found in polynomial time. The time complexity is $O(k^{\lceil \frac{1}{\varepsilon} \rceil} n^{\lceil \frac{1}{\varepsilon} \rceil + 1})$.*

Proof. For any fixed ρ, the number of possible i-tuples with $i \leq \rho$ is $O(n^\rho)$. In step 1.2, it takes $O((i+1)k|E|)$ time to find all hops shortest paths [11]. By lemma 2, step 1.3 takes $O(k^{i+1}n^2)$ time. Step 1.4 takes $O(n \log n)$ time to build the shortest paths forest. Clearly, step 1.5 and 1.6 can be done in $O(n \log n)$ time by any optimal sorting algorithm then to find $k - |V(X)|$ vertices to $\mathcal{L}_1$. Thus the time complexity of the algorithm is $O(k^{i+1}n^{\rho+2})$. By Lemma 6, the algorithm returns a $(1 + \frac{1}{\rho+1})$-approximation solution. For any desired ratio $(1 + \varepsilon)$, $0 < \varepsilon \leq 1$, we take $\rho = \lceil \frac{1}{\varepsilon} \rceil - 1$ and run the algorithm to obtain the desired approximation solution and the time complexity is $O(k^{\lceil \frac{1}{\varepsilon} \rceil} n^{\lceil \frac{1}{\varepsilon} \rceil + 1})$, which is polynomial for fixed ε. □

4 The 2-kBSRT Problem

In the section, we discuss the 2-kBSRT problem. In the 2-kBSRT problem, we are given a graph $G = (V, E, w)$ with two source vertices and asked to find a k-tree T. The objective function is $C_S(T) = \max_{s \in S}\{\sum_{v \in V(T)} d_T(s, v)\}$. We shall give a $(2 + \varepsilon)$-approximation algorithm for any fixed $\varepsilon > 0$. The algorithm is basically the same as algorithm **PTAS-2-kMRCT** except that, among all the constructed trees, it returns the tree T with minimum $C_S(T)$ instead of $C_R(T)$. The main idea is similar to the PTAS for the 2-kMRCT.

Lemma 7. *Let T be a k-tree of $G = (V, E, w)$ and P be the path between s_1 and s_2 on T. $C_S(T) \geq \frac{k}{2} w(P) + \sum_{v \in V(T)} d_T(v, P)$.*

Proof. For any $s \in S$, $D_T(s,V) = \sum_{v \in V(T)} \{d_T(v,P) + w(SP_T(v,s) \cap P)\}$, where $SP_T(v,s)$ is the path between v and s in T. Hence

$$\begin{aligned} C_S(T) &= \max\{D_T(s_1, V(T)), D_T(s_2, V(T))\} \\ &\geq \frac{1}{2}\{D_T(s_1, V(T)) + D_T(s_2, V(T))\} \\ &\geq \frac{k}{2} w(P) + \sum_{v \in V(T)} d_T(v,P). \end{aligned}$$

□

In the remaining paragraphs of this section, we shall use the following notations. Let Y be the optimal k-tree of the 2-kBSRT. Also let $P = (p_1 = s_1, p_2, p_3, \ldots, p_\lambda = s_2)$ be the path between s_1 and s_2 on Y. As defined in Section 3, let $M = \{m_0 = s_1, m_1, m_2, \ldots, m_\rho, m_{\rho+1} = s_2\}$ be a set of not necessarily distinct vertices of $V(P)$, which partitions $V(Y)$ into $U, U_0, U_1, \ldots, U_\rho$ such that $|U_i| \leq \frac{k}{\rho+1}$. The definitions of P_i, V_i are also the same as in the previous section. Our algorithm tries to correctly guess $m_1, m_2, \ldots, m_\rho$ and constructs a tree X spanning these vertices with the property that $d_X(v,s_1) + d_X(v,s_2) \leq w(P)$ for any $v \in V(X)$ and $|V(X)| \leq k$. Once such a tree X has been constructed, finding a shortest path forest F of G with multiple root $V(X)$. Let a spanning tree T' be the union of the forest F and X. For the tree T', we partition $V \setminus V(X)$ into two vertices sets $\mathcal{L}_1$ and $\mathcal{L}_2$ such that $|\mathcal{L}_1| = k - |V(X)|$, $|\mathcal{L}_2| = n - k$ and $d_G(l_1, X) \leq d_G(l_2, X)$ for any $l_1 \in \mathcal{L}_1$ and $l_2 \in \mathcal{L}_2$. The k-tree T is the subtree of T' induced by $\mathcal{L}_1 \cup V(X)$. We output the k-tree T with minimum $C_S(T)$ among all constructed trees. In the next lemma, we show that T is a $(2 + \frac{1}{\rho+1})$-approximation solution of the 2-kBSRT problem.

Lemma 8. *The k-tree T is a $(2 + \frac{1}{\rho+1})$-approximation solution of the 2-kBSRT problem.*

Proof. By definition, the vertices $m_1, m_2, \ldots, m_\rho$ partition the vertex set $V(Y)$ into $(U, U_0, U_1, \ldots, U_\rho)$ and the subsets satisfy the properties in Lemma 5. Since $d_X(v,s_1) + d_X(v,s_2) \leq w(P)$ for any $v \in V(X)$ with $|V(X)| \leq k$, we have

$$\begin{aligned} C_S(T) &= \max_{s \in S} \left\{ \sum_{v \in V(T)} d_T(v,s) \right\} \leq \sum_{v \in V(T)} d_T(v,X) + k\ w(P) \\ &= \sum_{v \in V(X)} d_T(v,X) + \sum_{v \in \mathcal{L}_1} d_T(v,X) + k\ w(P) \\ &= \sum_{v \in \{\mathcal{L}_1 \cap V(Y)\}} d_T(v,X) + \sum_{v \in \{\mathcal{L}_1 \setminus V(Y)\}} d_T(v,X) + k\ w(P). \end{aligned} \tag{7}$$

For each $v \in \{\mathcal{L}_1 \setminus V(Y)\}$, there exists a unique vertex $y \in \{V(Y) \setminus V(T)\} \subseteq \mathcal{L}_2$ such that $d_G(v,X) \leq d_G(y,X)$. The equation (7) will be

$$C_S(T) \leq \sum_{v \in V(Y)} d_G(v, X) + k\ w(P). \tag{8}$$

For $v \in U$, since $d_G(v, X) \leq d_G(v, M)$ and by Lemma 5, we have

$$\sum_{v \in U} d_G(v, X) \leq \sum_{v \in U} d_G(v, M) \leq \sum_{v \in U} d_Y(v, P). \tag{9}$$

Similarly $d_G(v, X) \leq d_G(v, M)$ for $v \in U_i$, and then by Lemma 5, we have

$$\begin{aligned} \sum_{0 \leq i \leq \rho} \sum_{v \in U_i} d_G(v, X) &\leq \sum_{0 \leq i \leq \rho} \left(\sum_{v \in U_i} d_Y(v, P) + \frac{k}{2(\rho+1)} w(P_i) \right) \\ &\leq \sum_{0 \leq i \leq \rho} \sum_{v \in U_i} d_Y(v, P) + \frac{k}{2(\rho+1)} w(P). \end{aligned} \tag{10}$$

By equations (8)–(10) and Lemma 7,

$$\begin{aligned} & C_S(T) \leq \sum_{v \in V(Y)} d_G(v, X) + k\ w(P) \\ & \leq \sum_{v \in U} d_Y(v, P) + \sum_{0 \leq i \leq \rho} \sum_{v \in U_i} d_Y(v, P) + \frac{k}{2(\rho+1)} w(P) + k\ w(P) \\ & = \sum_{v \in V(Y)} d_Y(v, P) + \frac{2\rho+3}{2(\rho+1)} k\ w(P) \\ & = \left(\sum_{v \in V(Y)} d_Y(v, P) + \frac{k}{2} w(P) \right) + \frac{\rho+2}{2(\rho+1)} k\ w(P) \\ & \leq C_S(Y) + \left(\frac{\rho+2}{\rho+1} \right) C_S(Y) = \left(2 + \frac{1}{\rho+1} \right) C_S(Y), \end{aligned}$$

since $C_S(Y) \geq \frac{k}{2} w(P)$. □

Theorem 2. *For any constant $\varepsilon > 0$, there is an algorithm finding a $(2+\varepsilon)$-approximation solution of the 2-kBSRT in $O(k^{\lceil \frac{1}{\varepsilon} \rceil} n^{\lceil \frac{1}{\varepsilon} \rceil + 1})$ time.*

Proof. By Lemma 8, we can find a $(2 + \frac{1}{\rho+1})$-approximation solution of the 2-kBSRT problem for any integer $\rho \geq 0$. Similar to Theorem 1, the time complexity is $O(k^{i+1} n^{\rho+2})$. For any $\varepsilon > 0$, we take $\rho = \lceil \frac{1}{\varepsilon} \rceil - 1$, and the result follows. □

5 The 2-kBVRT Problem

Finally, we present a PTAS for the 2-kBVRT problem in this section. The objective function of this problem is $C_V(T) = \max_{v \in V(T)} \{ \sum_{s \in \{s_1, s_2\}} d_T(s, v) \}$. The algorithm is basically the same as algorithm **PTAS-2-kMRCT** except that,

among all the constructed trees, it returns the k-tree T with minimum $C_V(T)$ instead of $C_R(T)$. The main idea is similar to the PTAS for the 2-kMRCT, but we need a different analysis of the performance.

The following property is useful for showing the performance of the algorithm.

Lemma 9. *Let T be any k-tree of $G = (V, E, w)$ and P be the path between s_1 and s_2 on T. $C_V(T) = w(P) + 2\max_{v \in V(T)}\{d_T(v, P)\}$.*

Proof. First, by definition, $D_T(v, S) = d_T(v, s_1) + d_T(v, s_2) = w(P) + 2d_T(v, P)$ for any $v \in V(T)$. Since $C_V(T)$ is the maximum $D_T(v, S)$ over all vertices in $V(T)$, the result follows. □

Let $\rho \geq 0$ be an integer parameter to be determined later. In the remaining paragraphs of this section, let Y be the optimal tree of the 2-kBVRT. Also let $P = (p_1 = s_1, p_2, p_3, \ldots, p_\lambda = s_2)$ be the path between the two sources on Y. The lemma comes from [4,5]. We state the result but the proof is omitted.

Lemma 10. *For any integer $\rho \geq 0$, there exists a subset $M \subset V(P)$ such that $|M| \leq \rho$ and $d_Y(v, M \cup \{s_1, s_2\}) \leq \frac{1}{\rho+2} w(P)$ for any $v \in V(P)$.*

The bound in the above lemma is tight. An extreme example is that the path have exactly $\rho + 1$ internal vertices and all the edges are of the same length. Since we can only choose ρ vertices, there is one vertex left and the distance to its neighbor is $w(P)/(\rho + 2)$. Let M be a vertex set satisfying the property in Lemma 10 and $\bar{M} = M \cup \{s_1, s_2\}$. The lemma comes from [4,5]. We state the result but the proof is omitted.

Lemma 11. *For any vertex $y \in V(Y)$, $d_G(y, \bar{M}) \leq d_Y(y, P) + \frac{1}{\rho+2} w(P)$.*

Let M be the set vertices in the i-tuple and $\bar{M} = M \cup \{s_1, s_2\}$, $i \leq \rho$. We show the result for the 2-kBVRT problem in the next lemma.

Lemma 12. *Let X be a tree spanning $\bar{M}$ and $d_X(v, s_1) + d_X(v, s_2) \leq w(P)$ for any $v \in V(X)$ with $|V(X)| \leq k$. Finding a shortest path forest F of G with multiple root $V(X)$. Let a spanning tree T' be the union of the forest F and X. For the tree T', we partition $V \setminus V(X)$ into two vertices sets $\mathcal{L}_1$ and $\mathcal{L}_2$ such that $|\mathcal{L}_1| = k - |V(X)|$, $|\mathcal{L}_2| = n - k$ and $d_G(l_1, X) \leq d_G(l_2, X)$ for any $l_1 \in \mathcal{L}_1$ and $l_2 \in \mathcal{L}_2$. The k-tree T, which is the subtree of T' induced by $\mathcal{L}_1 \cup V(X)$, is a $(1 + \frac{2}{\rho+2})$-approximation solution of the 2-kBVRT problem.*

Proof. For each vertex $v \in V(T)$, since it is connected to X by a shortest path, there exists a vertex $x \in V(X)$ such that $d_T(v, s_1) = d_G(v, x) + d_T(x, s_1)$ and $d_T(v, s_2) = d_G(v, x) + d_T(x, s_2)$. Therefore

$$D_T(v, S) = 2d_G(v, X) + d_T(x, s_1) + d_T(x, s_2) \leq 2d_G(v, X) + w(P).$$

For each vertex $v \in \{V(T) \setminus V(Y)\}$, there exists a unique vertex $y \in \{V(Y) \setminus V(T)\} \subseteq \mathcal{L}_2$ such that $d_G(v, X) \leq d_G(y, X)$, and hence we have $D_T(v, S) \leq 2d_G(v, X) + w(P) \leq 2d_G(y, X) + w(P)$. Since $\bar{M} \subset V(X)$, $d_G(y, X) \leq d_G(y, \bar{M})$,

and then by Lemma 11 we have $d_G(y, X) \leq d_Y(y, P) + \frac{1}{\rho+2}w(P)$. Consequently we have, for each vertex $v \in V(T)$, there exists a unique vertex $y \in V(Y)$ such that

$$\begin{aligned} D_T(v, S) &\leq 2d_G(y, X) + w(P) \leq 2d_Y(y, P) + \frac{2}{\rho+2}w(P) + w(P) \\ &= D_Y(y, S) + \frac{2}{\rho+2}w(P) \leq \left(1 + \frac{2}{\rho+2}\right) D_Y(y, S), \end{aligned}$$

since $w(P) \leq D_Y(y, S)$. Then by definition,

$$\begin{aligned} C_V(T) &= \max_{v \in V(T)} \{D_T(v, S)\} \leq \max_{v \in V(Y)} \{\left(1 + \frac{2}{\rho+2}\right) D_Y(v, S) \\ &= \left(1 + \frac{2}{\rho+2}\right) C_V(Y). \end{aligned}$$

□

Theorem 3. *For any constant $\varepsilon > 0$, there exists an algorithm, which finds a $(1+\varepsilon)$-approximation solution for the 2-kBVRT problem in $O(k^{\lceil \frac{2}{\varepsilon} \rceil - 1} n^{\lceil \frac{2}{\varepsilon} \rceil})$ time.*

Proof. By Lemma 12, there exists a $(1 + \frac{2}{\rho+2})$-approximation solution of the 2-kBVRT for any integer $\rho \geq 0$. Similar to Theorem 1, the time complexity is $O(k^{i+1} n^{\rho+2})$. For any $\varepsilon > 0$, we take $\rho = \lceil \frac{2}{\varepsilon} \rceil - 2$, and the result follows. □

6 Conclusion

In this paper, we investigate the 2-kMRCT, 2-kBSRT and 2-kBVRT problems and propose approximation algorithms. For the 2-kMRCT and 2-kBVRT problems, we show both problems admit PTASs. However, the 2-kBSRT seems more difficult to be approximated than the 2-kMRCT and the 2-kBVRT, we only have a $(2 + \varepsilon)$-approximation algorithm. An immediate open question is whether the approximation ratio can be improved to PTAS. Also, it would be interesting to see our techniques for the two sources to be generalized to the cases with more than two sources.

References

1. Arora, S., Karakostas, G.: A $2 + \varepsilon$ approximation algorithm for the k-MST problem. In: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2000), pp. 754–759 (2000)
2. Arya, S., Ramesh, H.: A 2.5-Factor approximation algorithm for the k-MST problem. Information Processing Letters 65, 117–118 (1998)
3. Blum, A., Ravi, R., Vempala, S.: A constant-factor approximation algorithm for the k-MST problem. Journal of Computer and System Sciences 58, 101–108 (1999)

4. Chen, Y.H., Wu, B.Y., Tang, C.Y.: Approximation algorithms for k-source bottleneck routing cost spanning tree problems. In: Laganà, A., Gavrilova, M., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) ICCSA 2004. LNCS, vol. 3044, pp. 355–366. Springer, Heidelberg (2004) (extended abstract)
5. Chen, Y.H., Wu, B.Y., Tang, C.Y.: Approximation algorithms for some k-source shortest paths spanning tree problems. Networks 47, 147–156 (2006)
6. Connamacher, H.S., Proskurowski, A.: The complexity of minimizing certain cost metrics for k-source spanning trees. Discrete Applied Mathematics 131, 113–127 (2003)
7. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithm, 2nd edn. MIT Press, Cambridge (2001)
8. Farley, A.M., Fragopoulou, P., Krumme, D.W., Proskurowski, A., Richards, D.: Multi-source spanning tree problems. Journal of Interconnection Networks 1, 61–71 (2000)
9. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, San Francisco (1979)
10. Garg, N.: A 3-approximation for the minimum tree spanning k vertices. In: Proceedings, 37th IEEE Symposium on Foundations of Computer Science (FOCS 1996), PP. 302–309 (1996)
11. Guérin, R., Orda, A.: Computing shortest paths for any number of hops. IEEE/ACM Transactions on Networking 10, 613–620 (2002)
12. Hu, T.C.: Optimum communication spanning trees. SIAM Journal on Computing 3, 188–195 (1974)
13. Johnson, D.S., Lenstra, J.K., Rinnooy, A.H.G.: The complexity of the network design problem. Networks 8, 279–285 (1978)
14. Ravi, R., Sundaram, R., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J.: Spanning trees short or small. SIAM Journal on Discrete Mathematics 9, 178–200 (1996)
15. Wu, B.Y., Chao, K.M., Tang, C.Y.: Approximation algorithms for some optimum communication spanning tree problems. Discrete Applied Mathematics 102, 245–266 (2000)
16. Wu, B.Y., Lancia, G., Bafna, V., Chao, K.M., Ravi, R., Tang, C.Y.: A polynomial time approximation scheme for minimum routing cost spanning trees. SIAM Journal on Computing 29, 761–778 (2000)
17. Wu, B.Y.: A polynomial time approximation scheme for the two-source minimum routing cost spanning trees. Journal of Algorithm 44, 359–378 (2002)
18. Wu, B.Y.: Approximation algorithms for the optimal p-source communication spanning tree. Discrete Applied Mathematics 143, 31–42 (2004)

On the Expected Value of a Number of Disconnected Pairs of Nodes in Unreliable Network*

Alexey S. Rodionov[1], Olga K. Rodionova[2], and Hyunseung Choo[3]

[1] Institute of Computational Mathematics and Mathematical Geophysics
SB RAS, Novosibirsk, Russia +383-3326949
alrod@rav.sscc.ru
[2] Siberian State University of Telecommunications and Informatics
Novosibirsk, Russia +383-2669343
rolcon@mail.ru
[3] School of Information and Communication Engineering
Sungkyunkwan University
440-746, Suwon, Korea +82-31-290-7145
choo@ece.skku.ac.kr

Abstract. The task of calculating the expectation of a number of disconnected pairs of nodes (EDP) in unreliable network is discussed. The task is NP-hard, that is it requires exhaustive search of subgraphs. The techniques for decreasing a number of enumerated subgraphs by using the branching (factoring) method and taking advantage from possible structural features are discussed. Usage of chains for reduction of the task dimension is considered.

1 Introduction

Random graphs are an acknowledged model for networks of different kinds. For the analysis of network's reliability the probability of connectivity (PoC) is used mostly ([1,2,3,4,5,6], for example) while the expectation of a number of disconnected pairs of nodes (EDP) is sometimes more valuable and informative index. Really, while PoC informs us about the possibility of an arbitrary node been connected with all other nodes of the network, EDP gives the information about *how many pairs of nodes can communicate in average.* For example all trees are equal from the point of the PoC while they are not from the point of EDP. Simple example is shown in fig.1. There are three 10-node trees of different configuration. If all edges are equally reliable (have the probability of existence p) then PoC is p^9 for all of them while EDPs for these trees are: a)

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment), IITA-2006-(C1090-0603-0046), and grant CS-9505.2006.9 of President of Russian Federation. Corresponding author: H. Choo.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 534–543, 2007.
© Springer-Verlag Berlin Heidelberg 2007

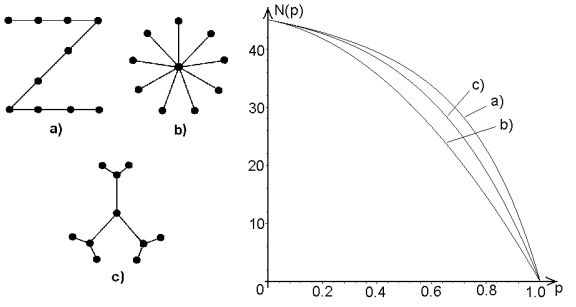


Fig. 1. Examples of 10-node trees and their EDP-polynomials

$45 - 9p - 8p^2 - 7p^3 - 6p^4 - 5p^5 - 4p^6 - 3p^7 - 2p^8 - p^9$; b) $45 - 9p - 36p^2$; c) $45 - 9p - 12p^2 - 12p^3 - 12p^4$. Plots of these polynomials are also presented in fig. 1. It is obvious that the tree in fig. 1.b) is preferable from the point of minimum EDP. Note that the maximal difference between EDPs for the cases a) and b) is 8.3245 in point 0.7472 while the whole number of pairs is 45.

Complementary index to EDP is a number of *connected* pairs of nodes (ECP), their sum is equal to the whole number of pairs of nodes in a network. Finding EDP requires exhaustive search of network destructions for its calculation. This is one of the reasons why few of researches deal with it. We can refer to the paper [7] where this index is mentioned among other valuable indexes of a network reliability. First we considered the problem in [8], where some preliminary results are presented. This paper contains some improvements and full proof of the main theorem in [1] that was unpublished, and presents main and absolutely new results concerning using long chains for reducing of the task dimension.

The rest of the paper is organized as follows: in Section 2 we give main definitions and notations. In Section 3 we give the exact equations for EDP for some kinds of graphs and in Section 4 we present equations for branching by chain. Case studies and brief conclusion are presented in the Sections 5 and 6.

2 Definitions and Notations

We consider random graphs with reliable nodes and unreliable edges. Let us denote:

$G(n, m) = (V, U, P, WT)$ – non-oriented graph with a set of nodes V, set of edges U, matrix of edges reliabilities P and vector of nodes weights WT;
$n = |V|$, $m = |U|$ – number of nodes and edges, respectively;
$w_i = w(v_i)$ – weight of a node v_i, $WT = w_1, \ldots, w_n$;
$W(G)$ – total weight of all nodes of G;
p_{ij} – probability of an edge e_{ij} being existent (being in a working state, edge's *reliability*), $P = ||p_{ij}||$; $q_{ij} = 1 - p_{ij}$;

$a_{ij}(G)$ – probability of an event that v_i and v_j are disconnected in G. If there is no variant reading, then we use simply a_{ij} without mentioning a graph's name;

$M(G,P)$ **and** $N(G,P)$ – ECP and EDP of a random graph G.

We use simply G for a graph if its n and m are clear from the context. If we need refer to a weight of i-th node in some graph G, then we use $WT_i(G)$. If we consider some special edges then we usually assign them personal numbers that is use notation e_k (k-th edge) instead of e_{ij} (an edge that connects v_i and v_j). Notation p_k is used for corresponding edge's reliability.

For simplifying some equations we assume that $\prod_{s=i+1}^{i} p_s \equiv 1$.

For shortening of expressions we use $S_m^k *$ for $\sum_{i=m}^{k} *$ and $S_k *$ for a $\sum_{i=1}^{k} *$.

A weight w_i that is equal to a number of nodes that had been contracted to form a special node v_i (initial weight of each node is 1) is needed for keeping the number of disconnected pairs in case when this node is separated from some other nodes in a graph.

We say that a random graph is connected (disconnected) when its structure is connected, not realization. In last case we say that "realization of a random graph is connected (disconnected)."

It is obvious that

$$N(G) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} a_{ij} w_i w_j \; . \tag{1}$$

The branching (factoring) method for calculating an expected value of any function of a random graph G is based on the equation of composite probability by two alternative hypothesis: existence or non-existence of some edge e_{ij}. Thus

$$N(G) = p_{ij} N(G_{ij}^*) + (1 - p_{ij}) N(G \backslash e_{ij}) \; , \tag{2}$$

where G_{ij}^* is a graph obtained from G by contracting v_i and v_j by an edge e_{ij}, $G \backslash e_{ij}$ – graph obtained from G by deleting the edge e_{ij}. Recursions go on until deriving a graph for which $N(G)$ can be obtained easily by an appropriate equation. For example, equations for EDP of a 2,3 or 4-node graphs, cycle, chain and a star-like tree are presented in [8].

3 Using Structural Peculiarities

First let us make the following derivation. During factoring process by equation (2) we may obtain several graphs with the same structure and matrix P but with different weights of nodes. In this case we can gain from the following useful lemmas and equations (proves are given in [8]).

Lemma 1. *If during a graph G factoring process some subgraphs $G_1, \ldots, G_k$ with same structure and matrix P are obtained in which only one special node v_s*

has different weight w_{si} in G_i, $i = 1, \ldots, k$, then the total contribution of these subgraphs into $N(G)$ is equal to $S_k(p_i) \cdot N(G^o)$, where p_i is a probability of G_i's realization, and graph G^o has the same structure and P as G_i and $WT_s(G^o) = S_k(p_i w_{si})/S_k(p_i)$.

Lemma 2. *If during a graph G factoring process some subgraphs $G_1, \ldots, G_k$ with same structure and matrix P are obtained in which v_s and v_t have weights w_{si} and w_{ti} in G_i, $i = 1, \ldots, k$, then the total contribution of these subgraphs into $N(G)$ is equal to $S_k(p_i) \cdot N(G^o) + \Delta$, where*

$$\Delta = a_{st} \frac{S_k(p_i) S_k(p_i w_{ti}) - S_k(p_i w_{si}) \cdot S_k(p_i w_{ti})}{S_k(p_i)} , \tag{3}$$

graph G^o has the same structure and matrix P as all G_i and

$$WT_s(G^o) = S_k(p_i w_{si})/S_k(p_i) ; \qquad WT_t(G^o) = S_k(p_i w_{ti})/S_k(p_i) . \tag{4}$$

k-component Graphs. If a graph G consist of k mutually disconnected subgraphs $G_1, \ldots, G_k$ then, obviously

$$N(G) = \sum_{i=1}^{k} N(G_i) + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} W(G_i) W(G_j) . \tag{5}$$

Deleting Dangling Nodes. Let a connected graph $G(n, m)$ have a dangling node v_i adjacent to some node v_j. Then

$$N(G) = N(G^o) + (1 - p_{ij}) w_i \left[W(G) - w_i \right] , \tag{6}$$

where G^o — a graph which has the same structure and a weight of v_j

$$WT_j(G^o) = p_{ij}(w_j + w_i) + (1 - p_{ij}) w_j = w_j + p_{ij} w_i . \tag{7}$$

4 Branching by Chain

4.1 Case of $k = 2$

In [8] we consider the case of 2-edge chains. The following theorem have been proven (unfortunately, we must admit that one item was omitted in the main equation, here is the right statement and improved proof).

Theorem 1. *If a connected random graph has a simple chain $\{C = e_{sx}, e_{xt}\}$ connecting nodes v_s and v_t through a node v_x with degree 2 then the following equation is true.*

$$\begin{aligned} N(G) = {} & [p_{st} + (1 - p_{st}) p_{sx} p_{xt}] N(G^*) + \big(1 - p_{st}\big)(1 - p_{sx} p_{xt}) N(G^o) - \\ & a_{st} \frac{(1 - p_{st})(1 - p_{xt})(1 - p_{sx}) p_{sx} p_{xt}}{(1 - p_{sx} p_{xt})} w_x^2 + \\ & (1 - p_{sx} - p_{xt} + p_{sx} p_{xt}) w_x \left[W(G) - w_x \right] , \end{aligned} \tag{8}$$

where

G^* is a graph, obtained from G by contracting v_s and v_t by the chain and removing e_{st} (if exists),
G^o is obtained from G by removing the chain and edge e_{st}.

Proof. Similar to [1] we make consequent factoring by states of edges and then collect terms. The process is shown in the fig. 2. If e_{st} exists, then we first choose it for factoring. Then we consequently make factoring by e_{sx} and e_{xt}. Two terminal graphs G_1 and G_4 we can easy replace by graphs with a structure of G_5 and G_6, correspondingly, using (6) and (7) (note that in G_1 we first replace multi-edge by a single one with equivalent reliability):

$$N(G_1) = N(G_5^0) + (1 - p_{xt} - p_{sx} + p_{xt}p_{sx})w_x\,[W(G) - w_x]\,, \tag{9}$$
$$WT(G_5^0, st) = w_s + w_t + (p_{sx} + p_{xt} - p_{sx}p_{xt})w_x\ ,$$
$$N(G_4) = N(G_6^0) + (1 - p_{xt})w_x\,[W(G) - w_x]\,, \quad WT(G_6^0, t) = w_t + p_{xt}w_x\ .$$

For graphs G_5 and G_6 themselves we have:

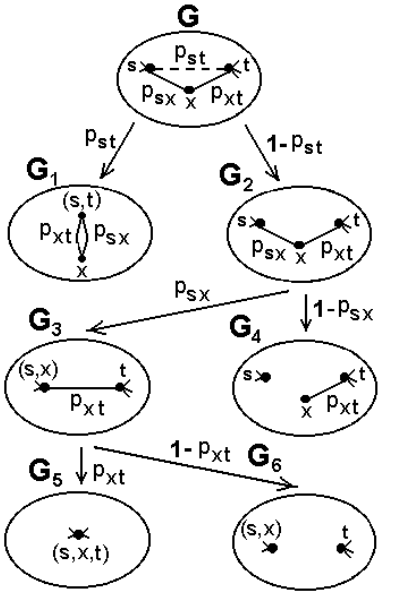


Fig. 2. Factoring in a graph with a 2-edge chain

$$WT(G_5, xst) = w_s + w_t + w_x\ , \tag{10}$$
$$WT(G_6, s) = w_s + w_x\ ,$$
$$WT(G_6, t) = w_t\ .$$

It is clear that graphs G_1, G_4, G_5 and G_6 are obtained with probabilities p_{st}, $(1 - p_{st})(1 - p_{sx})$, $(1 - p_{st})p_{sx}p_{xt}$ and $(1 - p_{st})(1 - p_{xt})p_{sx}$, correspondingly.

According to lemma 1 we can change calculation of EDP for G_5 and G_5^0 to calculation of EDP for some graph G^*, that has a structure of G_5 with a weight of joint node

$$WT(G^*, sxt) = w_s + w_t + \frac{(p_{sx}p_{st} + p_{sx}p_{xt} + p_{st}p_{xt} - 2p_{sx}p_{st}p_{xt})}{p_{st} + p_{sx}p_{st} - p_{sx}p_{st}p_{xt}} w_x \ . \quad (11)$$

For G_6 and G_6^0 we use lemma 2 as two nodes v_s and v_t have different weights in them. According to this lemma a total contribution of G_6 and G_6^0 into EDP of G is

$$[p_{st} + (1-p_{st})p_{sx}p_{xt}]N(G^*) + p_{st}(1 - p_{xt} - p_{sx} + p_{xt}p_{sx})w_x \left[W(G) - w_x\right] \ . \quad (12)$$

where G^o has a structure of G without chain e_{sx}, e_{xt} and edge e_{st} and

$$WT_s(G^o) = w_s + \frac{p_{sx}(1-p_{xt})}{1 - p_{sx}p_{xt}} w_x \ , \quad (13)$$
$$WT_t(G^o) = w_t + \frac{(1-p_{xs})p_{xt}}{1 - p_{sx}p_{xt}} w_x \ .$$

Now, using already obtained result for G^*, we have that

$$\begin{aligned} N(G) = {} & [p_{st} + (1-p_{st})p_{sx}p_{xt}]N(G^*) + \quad (14) \\ & p_{st}(1 - p_{xt} - p_{sx} + p_{xt}p_{sx})w_x \left[W(G) - w_x\right] + \\ & \big(1-p_{st}\big)\Big\{(1-p_{sx}p_{xt})\Big[N(G^o) - a_{st}\frac{(1-p_{xt})(1-p_{sx})p_{sx}p_{xt}}{(1-p_{sx}p_{xt})^2} w_x^2\Big] + \\ & (1-p_{xt})(1-p_{sx})w_x \left[W(G) - w_x\right]\Big\} \\ = {} & [p_{st} + (1-p_{st})p_{sx}p_{xt}]N(G^*) + \big(1-p_{st}\big)(1-p_{sx}p_{xt})N(G^o) - \\ & a_{st}\frac{(1-p_{st})(1-p_{xt})(1-p_{sx})p_{sx}p_{xt}}{(1-p_{sx}p_{xt})} w_x^2 + \\ & (1 - p_{sx} - p_{xt} + p_{sx}p_{xt})w_x \left[W(G) - w_x\right] \ , \end{aligned}$$

just as expected. □

Note that if there is no edge e_{st} in G, then we have simpler equation:

$$\begin{aligned} N(G) = {} & p_{sx}p_{xt}N(G^*) + (1 - p_{sx}p_{xt})N(G^o) - \quad (15) \\ & a_{st}\frac{(1-p_{xt})(1-p_{sx})p_{sx}p_{xt}}{(1-p_{sx}p_{xt})} w_x^2 + \\ & (1 - p_{sx} - p_{xt} + p_{sx}p_{xt})w_x \left[W(G) - w_x\right] \ . \end{aligned}$$

where $WT(G^*, sxt) = w_s + w_t + w_x$.

Now let us consider value of $a_{st} = a_{st}(G \backslash \{e_{st}, e_{sx}, e_{xt}\})$. There are well-known algorithms for finding 2-terminal probabilistic connectivity (see [6], for example) and a complexity of this task is obviously less then complexity of calculation of EDP for G_1 or G_4. Thus using equation (8) can be effective and this is shown further in the case studies.

4.2 Case of $k > 2$

Now let us present new theorem that allows us gain from possible existence of chains with more than 2 edges in a network.

Theorem 2. *Let a graph $G(n, m)$ have a chain of k edges $e_1 = (v_s(v_0), v_1)$, $e_2 = (v_1, v_2), \ldots, e_k = (v_{k-1}, v_t(v_k))$ with reliabilities p_1, $p_2, \ldots, p_k$, that connects nodes v_s and v_t with degrees more than 2. Then EDP for this graph can be represented in a form*

$$N(G) = \Big[p_{st} + (1 - p_{st}) \prod_{i=1}^{k} p_i\Big] \cdot N(G^*) + \Big[\sum_{i=0}^{k-1} (1 - p_{i+1}) \prod_{j=i+2}^{k} p_j\Big] \cdot N(G^o) + C \,, \tag{16}$$

where

- G^* *has a structure of a graph, obtained from G by contracting v_s and v_t by the chain and deleting e_{st} (if exists),*
- G^o *has a structure of a graph, obtained from G by deleting the chain and edge e_{st}.*

Item C and weights $WT_s(G^*)$, $WT_s(G^o)$ and $WT_t(G^o)$ are polynomial functions of w_i and p_{ij}, $i, j = 1, \ldots, n$.

For proving it we first prove the following lemmas.

Lemma 3. *If a graph G_k^- consists of a graph G' that is connected by a node v_s with a chain $C_k = \{e_1 = (v_s, v_1), \ldots, e_k = (v_{k-1}, v_k)\}$, that is ended by a dangling node, then*

$$N(G_k^-) = N(G^o) + \sum_{i=1}^{k} (1 - p_i) \Big[W(G') + \sum_{j=1}^{i-1} w_j\Big] \Big[w_i + \sum_{j=i+1}^{k} w_j \prod_{s=i+1}^{j} p_s\Big] \,, \tag{17}$$

where a graph G^o differs from G' by a weight of node v_s, that is equal to

$$WT_s(G^o) = w_s^o = w_s + \sum_{i=1}^{k} w_i \prod_{j=1}^{i} p_j \,. \tag{18}$$

Proof is based on the consequential applying of equations (6) and (7) to a chain's nodes, starting from the last one. Thus

$$N(G_k^-) = N(G_{k-1}^{-o}) + (1 - p_k) w_k \left[W(G_k^-) - w_k\right] = \ldots \tag{19}$$

$$= N(G_0^{-o}) + \sum_{i=1}^{k} (1 - p_i) w_i^o \left[W(G_i^{-o}) - w_i^o\right] \,, \tag{20}$$

where as $N(G_i^{-o})$ we designate an intermediate graph with attached chain of a length i. It is simple to obtain that $w_i^o = w_i + \sum\limits_{j=i+1}^{k} w_j \prod\limits_{s=i+1}^{j} p_s$, and $W(G_i^{-o}) - w_i^o = W(G') + \sum\limits_{j=1}^{i-1} w_j$, from which we have a desired. □

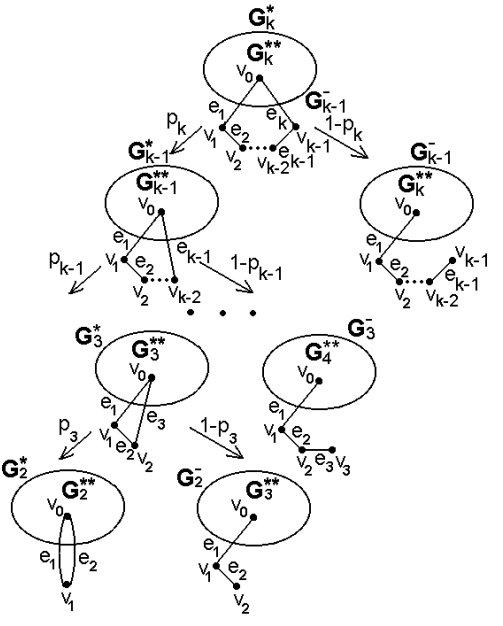


Fig. 3. Factoring with a graph that contains an attached cycle

Lemma 4. *Let G_k^* be a graph that has a cyclical block with k edges*

$$e_1 = (v_0(v_s), v_1), e_2 = (v_1, v_2), \ldots, e_i = (v_{i-1}, v_i), \ldots, e_k = (v_{k-1}, v_0(v_s)) \,,$$

that have reliabilities $p_1,\ p_2, \ldots, p_k$. Let $G' = G \backslash \{e_1, \ldots, e_k, e_{st}\}$. Then

$$N(G_k^*) = N(G^{o*})\Big[\prod_{i=3}^{k} p_i + \sum_{i=3}^{k-1}(1-p_{i+1}) \prod_{j=i+2}^{k} p_i\Big] + \tag{21}$$

$$(1-p_1-p_2+p_1p_2)w_1\big[W(G')+S_2^k(w_i)\big] +$$

$$\sum_{i=3}^{k-1}\sum_{j=1}^{i}(1-p_j)\Big[W(G')+\sum_{s=i+1}^{k-1} w_s+\sum_{s=1}^{j-1} w_s\Big]\Big[w_j+\sum_{s=j+1}^{k-1} w_s \prod_{\nu=j+1}^{s} p_\nu\Big] \,,$$

$$WT_0(G^*) = \Big[w_0 + S_3^{k-1}(w_i) + (p_1 + p_2 - p_1p_2)w_1\Big] \prod_{i=3}^{k} p_i + \tag{22}$$

$$\sum_{i=3}^{k-1}\Big[(1-p_{i+1}) \prod_{j=i+2}^{k} p_i\Big]\Big(w_0 + \sum_{j=i+1}^{k} w_j + \sum_{j=1}^{i} w_j \prod_{s=1}^{j} p_s\Big) \,.$$

Proof is based on consequent branching by states of edges of the cycle as is shown in fig. 3 and using previous lemmas: graph G_2^* is a graph with a dangling node, graphs $G_i^{\sim}$ satisfy conditions of the lemma 3. After applying lemma 1 to resulting graphs we obtain which required. □

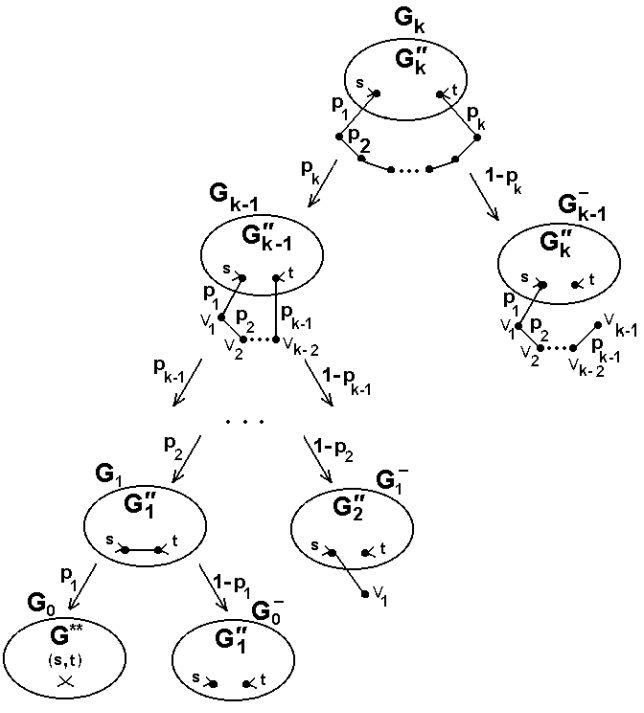


Fig. 4. Factoring in a graph with long chain

For proving the theorem we first make factoring by state of an edge e_{st}, that encloses the chain With a probability p_{st} we obtain a graph G_k^* by contracting branch (let us keep number 0 for the joined node) that satisfies conditions of the lemma 4, while with a probability $1 - p_{st}$ by deleting branch we obtain a graph $G \backslash e_{st}$ that can be treated as is shown in fig. 4. After applying lemma 3 to all intermediate graphs with "attached chains" and using lemmas 1 and 2 we obtain which required. □

5 Case Studies

As there are no other known algorithms for the task (except exhaustive search) we compare factoring method, factoring method with removing dangling nodes (FMRDN) and factoring method with removing dangling nodes and reducing of chains (FMRC).

Three graphs were chosen for testing: G_1, G_2 – the lattices 4x3 (12 nodes, 17 edges) and 4x4 (16 nodes, 24 edges), correspondingly; and G_3 – wheel-like cubic graph (cycle with chords that connect opposite nodes) with 12 nodes and 18 edges. Computer Intel Celeron 2.2GHz processor, 256Mb RAM was used. It is clear that using of our technique allows considerable speed up in calculations.

Table 1. Experimental results

Graph	Number of recursions / Calculation time (msec)		
	Factoring method	FMRDN	FMRC
G_1	82611/48110	2323/219	194/109
G_2	$> 10^7 /> 1$ hour	37677/39484	2019/12391
G_3	82152/53703	7350/343	993/266

6 Conclusion

Thus we have presented some useful equations that can help in calculating the EDP of a random graph. Most advantage for speeding up is gained from branching by chains. Note that chains are inevitable during the factoring process as a result of edge deletion and, sometimes, as a result of contracting by edge (refer to [1]). We hope that this paper will induce interest to the analysis of EDP and its calculation. Our current researches concern the approximate calculation of EDP and we intend present first correspondent results in nearest future.

References

1. Rodionova, O.K., Rodionov, A.S., Choo, H.: Network Probabilistic Connectivity: Exact Calculation with Use of Chains. In: Laganà, A., Gavrilova, M., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) ICCSA 2004. LNCS, vol. 3046, pp. 315–324. Springer, Heidelberg (2004)
2. Krivoulets, V.G., Polesskii, V.P: What is the Theory of Bounds for Network Reliability? Inf. Processes 1(2), 199–203 (2001)
3. Cancela, H., Petingi, L.: Diameter constrained network reliability: exact evaluation by factorization and bounds. In: Proc. of the Int. Conf. on Industrial Logistics, Japan, pp. 359–366 (2001)
4. Lucet, C., Manouvrier, J.-F.: Exact Methods to compute Network Reliability. In: Proc. of 1st International Conf. on Mathematical Methods in Reliability, Bucharest, Roumanie (September 1997), paper available via `http://ramp.ucsd.edu/resources.html`
5. Satyanarayana, A., Chang, M.K.: Network reliability and the factoring theorem. Networks 13, 107–120 (1983)
6. Shooman, A.M.: Algorithms for Network Reliability and Connection Availability Analysis. In: Electro/95 Int. Professional Program Proc., pp. 309–333 (1995)
7. Palmer, C.R., Siganos, G., Faloutsos, M., Faloutsos, C., Gibbons, P.: The connectivity and faulttolerance of the Internet topology. Workshop on Network-Related Data Management (NRDM-2001), paper available via `http://www.research.att.com/~divesh/papers/cjfgs98-ir.ps`
8. Rodionov, A., Rodionova, O.: Network Probabilistic Connectivity: Expectation of a Number of Disconnected Pairs of Nodes. In: Gerndt, M., Kranzlmüller, D. (eds.) HPCC 2006. LNCS, vol. 4208, pp. 101–109. Springer, Heidelberg (2006)
9. Shooman, A.M., Kershenbaum, A.: Exact Graph-Reduction Algorithms for Network Reliability Analysis. In: Proc. GLOBECOM' 91. vol. 2, pp. 1412–1420 (1991)

Linearization of Stream Ciphers by Means of Concatenated Automata*

A. Fúster-Sabater[1] and P. Caballero-Gil[2]

[1] Instituto de Física Aplicada, C.S.I.C.
Serrano 144, 28006 Madrid, Spain
amparo@iec.csic.es
[2] DEIOC, University of La Laguna
38271 La Laguna, Tenerife, Spain
pcaballe@ull.es

Abstract. Binary sequences generated by a class of linear cellular automata (multiplicative polynomial cellular automata) can be written as solutions of linear difference equations. At the same time, such solutions equal the output sequences from well known LFSR-based stream ciphers. These simple linear automata easily model complex nonlinear cryptographic generators. The modelling technique is based on the concatenation of basic 90/150 cellular automata. Some illustrative examples covering a wide range of cryptographic generators complete the work.

Keywords: Stream cipher, cellular automata, concatenation, symmetric cryptography.

1 Introduction

Confidentiality makes use of an encryption function currently called *cipher* that converts the *plaintext* into the *ciphertext*. Ciphers are usually divided into two large classes: stream ciphers and block-ciphers. Stream ciphers are very fast (in fact, the fastest among the encryption procedures) so they are implemented in many technological applications e.g. algorithms A5 in GSM communications [10] or the encryption system E0 used in the Bluetooth specifications [1]. Stream ciphers try to imitate the ultimate one-time pad cipher and are supposed to be good pseudorandom generators capable of stretching a short secret seed (the secret key) into a long sequence of seemingly random bits (the keystream sequence). This sequence is then XORed with the plaintext in order to obtain the ciphertext.

Most keystream generators are based on Linear Feedback Shift Registers (LFSRs) [8] whose output sequences are combined in a nonlinear way. Combinational generators, nonlinear filters, clock-controlled generators, multi-speed generators are just some of the most popular sequence generators. All these structures produce keystreams with high linear complexity, long period and good statistical properties (see [5] and [2]).

* Work supported by Ministerio de Educación y Ciencia (Spain) Projects SEG2004-02418 and SEG2004-04352-C04-03.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 544–554, 2007.
© Springer-Verlag Berlin Heidelberg 2007

On the other hand, bit sequences generated by one-dimensional binary Cellular Automata (CA) [3] have been found to generate exactly the same PN-sequences as those of the LFSRs. Nevertheless, in contrast with the LFSRs, the bit sequences generated by adjacent cells in CA are not correlated [15]. In this sense, linear CA are superior to common LFSRs. This is the reason why they are currently used in testing of digital circuits, Built-In Self-Test (BIST) schemes and self-checking. Moreover in [16], it is proved that linear CA are isomorphic to conventional LFSRs. Thus, the latter structures may be simply substituted by the former ones in order to accomplish the same goal: generation of keystreams with cryptographic application.

Nevertheless, the great advantage of above mentioned CA is that multiple generators designed in terms of LFSRs as nonlinear structures preserve the linearity when they are expressed under the form of CA. The question that arises in a natural way is: are there one-dimensional linear CA able to produce the sequence obtained from any LFSR-based nonlinear generator? The answer is yes and, in fact, this paper tackles the problem of given a particular LFSR-based generator how to find one-dimensional CA that reproduce exactly its output sequence. More precisely, in this work it is shown that wide classes of LFSR-based nonlinear generators, Clock-Controlled generators, Cascade-Clock-Controlled generators, Shrinking generator or the generators producing Kasami sequences, GMW sequences, No sequences, Klapper sequences ... (see [9] and the references cited therein), can be modelled in terms of linear CA.

All the sequences mentioned above are included in the class of *interleaved sequences* [9], that is pseudorandom sequences satisfying a common property: each sequence can be decomposed into a collection of shifts of an unique PN-sequence and zero sequences. This kind of sequences can be obtained from linear multiplicative polynomial CA. That is to say, CA made out of a basic structure concatenated a number of times. Moreover, the basic structure is a linear automaton based on rules 90 and 150, see [3]. Therefore, the goal of this work is double:

- To analyze linear multiplicative polynomial CA with emphasis on the cryptographic parameters of their generated sequences (period, linear complexity, characteristic polynomial, number of different output sequences etc.).
- To model the above nonlinear LFSR-based generators in terms of linear multiplicative polynomial CA.

Once the generators have been linearized, all the theoretical background on linear CA found in the literature can be applied to their analysis and/or cryptanalysis.

2 Fundamentals and Basic Notation

Our attention is concentrated on one-dimensional binary hybrid linear CA with three site neighborhood and rules 90 and 150 [12]. Both rules can be defined as follows:

$$\text{Rule 90} \qquad x_{n+1}^k = x_n^{k-1} + x_n^{k+1}$$

$$\text{Rule 150} \qquad x_{n+1}^k = x_n^{k-1} + x_n^k + x_n^{k+1}$$

Indeed, x_{n+1}^k the content of the *k-th* cell at time $n+1$ depends on the content of either two different cells (rule 90) or three different cells (rule 150) at time n, for $(k = 1, ..., L)$ where L is the length of the automaton. For a cellular automaton of length $L = 10$ cells, configuration rules $(90, 150, 150, 150, 90, 90, 150, 150, 150, 90)$ and initial state $(0, 0, 0, 1, 1, 1, 0, 1, 1, 0)$, Table 1 illustrates the behavior of this structure: the formation of its output sequences $\{x_n^k\}$ $(k = 1, ..., 10)$ (binary sequences read vertically) as well as the succession of states (binary configurations of 10 bits read horizontally). After a succession of 62 states, this automaton goes back to the initial state. All sequences $\{x_n^k\}$ in a state cycle have the same period, linear complexity and characteristic polynomial.

Table 1. A linear 90/150 cellular automaton of 10 cells

90	150	150	150	90	90	150	150	150	90
0	0	0	1	1	1	0	1	1	0
0	0	1	0	0	1	0	0	0	1
0	1	1	1	1	0	1	0	1	0
1	0	1	1	1	0	1	0	1	1
0	0	0	1	1	0	1	0	0	1
0	0	1	0	1	0	1	1	1	0
0	1	1	0	0	0	0	1	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	0	1	1	0	1	0	0
$\{x_n^1\}$	$\{x_n^2\}$	$\{x_n^3\}$	$\{x_n^4\}$	$\{x_n^5\}$	$\{x_n^6\}$	$\{x_n^7\}$	$\{x_n^8\}$	$\{x_n^9\}$	$\{x_n^{10}\}$

Next, some important definitions are introduced.

Definition 1. *A Multiplicative Polynomial Cellular Automaton is defined as a cellular automaton whose characteristic polynomial is of the form* $P_M(X) = (P(X))^p$ *where p is a positive integer. If* $P(X)$ *is a primitive polynomial, then the automaton is called a Primitive Multiplicative Polynomial Cellular Automaton.*

A widely accepted measure of the unpredictability of a sequence is its *linear complexity* [14] that can be defined as follows:

Definition 2. *The linear complexity of a sequence is the shortest linear recursion satisfied by such a sequence.*

For a binary sequence $\{x_n\}$, the linear recurrence relationship that specifies its *n-th* element can be written:

$$x_n + \sum_{i=1}^{r} c_i \, x_{n-i} = 0, \qquad n \geq r \tag{1}$$

where the sequence elements (x_n) as well as the coefficients (c_i) belong to the Galois Field $GF(2)$. The linear recursion can be expressed as a linear difference equation:

$$(E^r + \sum_{i=1}^{r} c_i \, E^{r-i}) \, x_n = 0, \qquad n \geq 0 \tag{2}$$

where E is the shifting operator that operates on x_n, i.e. $Ex_n = x_{n+1}$. The characteristic polynomial of the difference equation (2) is:

$$P(X) = X^r + \sum_{i=1}^{r} c_i \, X^{r-i}. \tag{3}$$

Let $P(X)$ be a primitive polynomial of degree r and $\alpha \in GF(2^r)$ one of its roots. In this case [13],

$$\alpha, \; \alpha^2, \; \alpha^{2^2}, \ldots, \; \alpha^{2^{(r-1)}} \tag{4}$$

are the r roots of such a polynomial. If $P_M(X) = (P(X))^p$, then the roots of $P_M(X)$ will be the same as those of $P(X)$ but with multiplicity p. Consequently, all the binary sequences generated by a primitive multiplicative polynomial cellular automaton will satisfy the linear difference equation:

$$(E^r + \sum_{i=1}^{r} c_i \, E^{r-i})^p \, x_n = 0, \qquad n \geq 0. \tag{5}$$

Our analysis focuses on all the possible solutions $\{x_n\}$ of this equation. For a particular sequence generated at the *k-th* cell, the notation will be $\{x_n^k\}$.

3 Structural Properties of Multiplicative Polynomial CA

In this section, the characteristics of the sequences obtained from this kind of CA are considered in terms of their cryptographic parameters (period, linear complexity and number of different sequences).

3.1 Period of the Generated Sequences

The solutions of the equation (5) are linear combinations of $p \cdot r$ solutions of the form [6]:

$$x_n = \sum_{i=0}^{p-1} \left(\binom{n}{i} \sum_{j=0}^{r-1} A_i^{2^j} \; \alpha^{2^j n} \right), \tag{6}$$

where $S_n^i = \sum_{j=0}^{r-1} A_i^{2^j} \; \alpha^{2^j n}$ represents the *n-th* element of a PN-sequence [13] of period $2^r - 1$ where $A_i \in GF(2^r)$. Thus, the binary sequence $\{x_n\}$ can be written as the sum of p times the same PN-sequence weighted by a binomial coefficient

$$\{x_n\} = \sum_{i=0}^{p-1} \binom{n}{i} \{S_n^i\}. \tag{7}$$

In addition, each binomial coefficient defines a succession of binary values with a constant period p_i. Therefore, the sequence $\{x_n\}$ is the sum of p sequences of distinct periods $T_i = p_i \cdot (2^r - 1)$ and the period of the total sequence will be:

$$T = \max\{T_i \ (i = 0, \ldots, p-1) \ / \ A_i \neq 0\}. \tag{8}$$

It can be noticed that the period of the different sequences $\{x_n\}$ generated by a multiplicative polynomial cellular automaton depends on the choice of the coefficients A_i in the equation (6). All the sequence generated at the same state cycle have the same period. Nevertheless, the same automaton can produce sequences with distinct periods depending on the state cycle considered.

3.2 Linear Complexity of the Generated Sequences

According to [13], the linear complexity of a sequence equals the number and multiplicity of characteristic polynomial roots that appears in the linear recurrence relationship. Therefore, coming back to the equation (6) the linear complexity of $\{x_n\}$ can be computed. In fact, we have r roots each of them with multiplicity p. Thus, if i_{max} is the greatest value of i $(i = 0, \ldots, p-1)$ for which $A_i \neq 0$, then the linear complexity LC of the sequence $\{x_n\}$ will be:

$$LC = (i_{max} + 1) \cdot r \ . \tag{9}$$

It can be noticed that the linear complexity of the different sequences $\{x_n\}$ generated by a multiplicative polynomial cellular automaton depends on the choice of the coefficients A_i in (6). All the sequence generated at the same state cycle have the same linear complexity. Nevertheless, the same automaton can produce sequences with distinct linear complexities depending on the state cycle considered.

3.3 The Number of Different Generated Sequences

In order to count the number of different sequences $\{x_n\}$ generated, the choice of the coefficients A_i in the equation (6) must be considered. Different situations can take place:

- If $A_i = 0 \ \ \forall \, i$, then all the cells of the cellular automaton will generate the identically null sequence.
- If $A_0 \neq 0$ and $A_i = 0 \ \ \forall \, i > 0$, then all the cells of the cellular automaton will generate a unique PN-sequence $\{S_n^0\}$ of period $T_0 = 2^r - 1$ and characteristic polynomial $P(X)$.
- If $A_0 \in GF(2^r)$, $A_1 \neq 0$ and $A_i = 0 \ \ \forall \, i > 1$, then there will be $2^r \cdot (2^r - 1)$ possible choices of (A_0, A_1). According to subsection (3.1), the period of these sequences will be $T_1 = p_1 \cdot (2^r - 1)$. Thus, the number of different sequences for these values of A_i is:

$$N_1 = \frac{2^r \cdot (2^r - 1)}{p_1 \cdot (2^r - 1)} = 2^{r-1}. \tag{10}$$

– In general, if $A_0, A_1, \ldots, A_{i-1} \in GF(2^r)$, $A_i \neq 0$ and $A_j = 0 \ \forall\, j > i$, then there will be $2^{(i\,r)} \cdot (2^r - 1)$ possible choices of $(A_0, A_1, \ldots, A_i)$. According to subsection (3.1), the period of these sequences will be $T_i = p_i \cdot (2^r - 1)$. Thus, the number of different sequences for these values of A_i is:

$$N_i = \frac{2^{(i\,r)} \cdot (2^r - 1)}{T_i} = \frac{2^{(i\,r)}}{p_i}. \tag{11}$$

The total number of different sequences obtained from a multiplicative polynomial cellular automaton will be:

$$N_{total} = \sum_{i=0}^{p-1} N_i. \tag{12}$$

In this computation the null sequence is excluded.

4 Linear Modelling by Concatenation of Primitive CA

In the previous section, structural properties of the sequences obtained from multiplicative polynomial cellular automata have been considered. Now the particular form of these automata is analyzed. In order to simplify the notation, CA we are dealing with can be represented by means of binary strings with the following codification: 0 = rule 90 and 1 = rule 150. Since the characteristic polynomial of these automata is $P_M(X) = (P(X))^p$, it seems quite natural to construct a multiplicative polynomial cellular automaton by concatenating p times the basic automaton (or the reverse version). In this way, the construction of a linear model based on CA is carried out by the generic algorithm:
Input: The parameters of a nonlinear keystream generator.

– *Step 1:* Determine the irreducible factor $P(X)$ of the characteristic polynomial of each interleaved sequence.
– *Step 2:* Compute the pair of basic CA whose characteristic polynomial is the irreducible factor $P(X)$.
– *Step 3:* For each one of these basic CA, construct by successive concatenations the longer cellular automaton able to generate the original interleaved sequence.

Output: Two linear CA producing the corresponding keystream sequence.

4.1 The Shrinking Generator

The shrinking generator [4] is a typical example of cryptographic generator with characteristic polynomial $P_M(X)$. The characteristics of this generator can be summarized as follows: The generator is made of two LFSRs, SR1 and SR2, with lengths $L_j (j = 1, 2)$ and feedback polynomials $P_j(X) (j = 1, 2)$, respectively. The decimation rule is: the bit produced by SR2 is discarded if the corresponding

bit of SR1 equals 0. The decimated sequence is just the output sequence of the generator. The generator characteristic polynomial is of the form $P_M(X) = (P(X))^p$, $P(X)$ being a primitive polynomial of degree $r = L_2$ and $2^{(L_1-2)} < p \leq 2^{(L_1-1)}$. Moreover, $P(X)$ is the characteristic polynomial of the cyclotomic *coset* E in $GF(2^{L_2})$ with $E = 2^0 + 2^1 + \ldots + 2^{L_1-1}$, see [8].

The output sequence will be a solution of a linear difference equation corresponding to primitive multiplicative polynomial CA. Consequently, the shrinking generator can be expressed in terms of a lineal model based on CA. A simple example illustrates the modelling procedure.

Input: A shrinking generator characterized by two LFSRs of lengths $L_1 = 3$, $L_2 = 5$ respectively and characteristic polynomial $P_2(X) = X^5+X^4+X^2+X+1$. Now, $p = 2^{L_1-1} = 4$ and $r = L_2 = 5$.

Step 1: $P(X)$ the irreducible factor of the generator characteristic polynomial is:

$$P(X) = X^5 + X^2 + 1 \; .$$

Step 2: The pair of basic CA whose characteristic polynomial is $P(X)$ are:

$$\begin{array}{l} \Delta_5 = 0\,1\,1\,1\,1 \\ \Delta_5^* = 1\,1\,1\,1\,0 \end{array}$$

Step 3: Computation of the required pair of CA by successive concatenations. For the first automaton:

$$\begin{array}{l} \Delta_5 = \;\; 0\,1\,1\,1\,1 \\ \Delta_{10} = 0\,1\,1\,1\,0\,0\,1\,1\,1\,0 \\ \Delta_{20} = 0\,1\,1\,1\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1\,0\,0\,1\,1\,1\,0 \text{ (final automaton)} \end{array}$$

For the second automaton:

$$\begin{array}{l} \Delta_5^* = \;\; 1\,1\,1\,1\,0 \\ \Delta_{10}^* = 1\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\ \Delta_{20}^* = 1\,1\,1\,1\,1\,1\,1\,1\,1\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1\,1 \text{ (final automaton)} \end{array}$$

For each automaton, the procedure in *Step 3* has been carried out $L_1 - 1$ times. In fact, each basic automaton with complementations has been concatenated $p = 2^{L_1-1}$ times.

Output: Two binary strings codifying the required CA.

In this way, we have obtained two linear CA:

(90,150,150,150,90,90,150,150,150,150,150,150,150,150,90,90,150, 150,150,90) and (150,150,150,150,150,150,150,150,150,150,90, 90,150,150 ,150,150,150,150,150, 150,150) both of them able to generate the shrunken sequence corresponding to the given shrinking generator. Consequently, the shrinking generator can be expressed in terms of lineal models based on CA.

4.2 The Class of Clock-Controlled Shrinking Generators

The Clock-Controlled Shrinking Generators (CCSGs) constitute a wide class of clock-controlled sequence generators [11] with applications in cryptography, error correcting codes and digital signature. An CCSG is a sequence generator composed of two LFSRs notated R_1 and R_2. The parameters of both registers are defined as those of subsection 4.1. At any time t, R_1 (the control register) is clocked normally while the second register R_2 is clocked a number of times given by an integer decimation function notated X_t. In fact, if $A_0(t)$, $A_1(t)$, ..., $A_{L_1-1}(t)$ are the cell contents of R_1 at time t, then X_t is defined as

$$X_t = 1 + 2^0 A_{i_0}(t) + 2^1 A_{i_1}(t) + \ldots + 2^{w-1} A_{i_{w-1}}(t) \tag{13}$$

where $i_0, i_1, \ldots, i_{w-1} \in \{0, 1, \ldots, L_1 - 1\}$ and $0 < w \leq L_1 - 1$.

In this way, the output sequence of a CCSG is obtained from a double decimation. First, $\{b_i\}$ the output sequence of R_2 is decimated by means of X_t giving rise to the sequence $\{b'_i\}$. Then, the same decimation rule P, defined in subsection 4.1, is applied to the sequence $\{b'_i\}$. Remark that if $X_t \equiv 1$ (no cells are selected in R_1), then the proposed generator is just the shrinking generator.

The generator characteristic polynomial [11] is of the form $P_M(X) = (P(X))^p$, $P(X)$ being a primitive polynomial of degree $r = L_2$ and $2^{(L_1-2)} < p \leq 2^{(L_1-1)}$. In fact, $P(X)$ depends on the characteristic polynomial of the register R_2, the length L_1 of the register R_1 and the decimation function X_t. Moreover, $P(X)$ is the characteristic polynomial of the *cyclotomic coset* E, where E is given by:

$$E = 2^{L_1-w} \left(\sum_{i=1}^{2^w} i\right) - 1 = (1 + 2^w)\, 2^{L_1-1} - 1. \tag{14}$$

According to the previous characteristics, the CCSG output sequence will be a solution of a linear difference equation corresponding to primitive multiplicative polynomial CA. Consequently, every CCSG can be expressed in terms of a lineal model based on CA. A simple example illustrates the modelling procedure.

Input: A CCSG characterized by two LFSRs R_1 of length $L_1 = 3$ and then R_2 of length $L_2 = 5$ and characteristic polynomial $P_2(X) = X^5 + X^4 + X^2 + X + 1$. The decimation function is $X_t = 1 + 2^0 A_0(t) + 2^1 A_1(t) + 2^2 A_2(t)$ with $w = 3$.

Step 1: $P(X)$ is the characteristic polynomial of the cyclotomic *coset E*. Now $E \equiv 4$ *mod* 31, that is we are dealing with the cyclotomic coset 1. Thus, the corresponding characteristic polynomial is:

$$P(X) = X^5 + X^4 + X^2 + X + 1\,.$$

Step 2: From $P(X)$ and applying the Cattell and Muzio synthesis algorithm [3], two linear CA whose characteristic polynomial is $P(X)$ can be determined. Such CA are written in binary format as:

$$\Delta_5 = 1\ 0\ 0\ 0\ 0$$
$$\Delta_5^* = 0\ 0\ 0\ 0\ 1$$

Step 3: Computation of the required pair of CA.
For the first automaton:

$$\begin{aligned}
\Delta_5 &= 1\,0\,0\,0\,0\\
\Delta_{10} &= 1\,0\,0\,0\,1\,1\,0\,0\,0\,1\\
\Delta_{20} &= 1\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,1 \quad \text{(final automaton)}
\end{aligned}$$

For the second automaton:

$$\begin{aligned}
\Delta_5^* &= 0\,0\,0\,0\,1\\
\Delta_{10}^* &= 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\\
\Delta_{20}^* &= 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0 \quad \text{(final automaton)}
\end{aligned}$$

For each automaton, the procedure in *Step 3* has been carried out twice as $L_1 - 1 = 2$.

Output: Two binary strings of length $L = 20$ codifying the required CA.

From a point of view of the CA-based linear models, the shrinking generator or any one of the CCGS are entirely analogous. Thus, the fact of introduce an additional decimation function does neither increase the complexity of the generator nor improve its resistance against cryptanalytic attacks since both kinds of generators can be linearized by the same class of CA-based models.

4.3 The Cascade Gollmann Generator

This generator [7] is made out of m cascaded LFSRs with primitive characteristic polynomials of degree n. In fact, the i-th register is controlled by all the previous registers j with $j < i$ in a stop-and-go way. The period of the generated sequence is $T = (2^n - 1)^m$ and the linear complexity is lower bounded by $LC > n(2^n - 1)^{m-1}$. For a very simple example of this generator with $m = 2$ LFSRs with primitive characteristic polynomials of degree $n = 3$, the output sequence is depicted in Table 1. Indeed, the generated sequence of period $T = 49$ can be read horizontally. In addition, every vertical column corresponds to shifts of a PN-sequence of characteristic polynomial $P(X) = X^3 + X^2 + 1$ or its complementary sequence.

In order to model this kind of generator, the basic automaton to be concatenated is made out of the automaton associated to $P(X)$ plus an additional cell due to the complementarity. The rest of the modelling procedure is analogous to those described in the previous examples.

5 Conclusions

In this work, structural properties of the primitive multiplicative polynomial CA have been analyzed. It is shown that wide classes of LFSR-based sequence generators with cryptographic application can be described in terms of CA-based structures. In this way, sequence generators conceived and designed as complex nonlinear models can be written in terms of simple linear models. The algorithm

Table 2. Horizontally: Output sequence of a cascade Gollmann generator. In vertical, different shifts of a PN-sequence or its complementary sequence.

$\overline{PN}$	PN	$\overline{PN}$	PN	PN	PN	$\overline{PN}$
1	1	0	0	0	0	1
1	1	0	1	1	1	0
0	0	1	1	1	1	0
1	0	1	1	1	1	0
0	1	0	0	0	0	1
0	0	1	0	0	0	1
0	1	0	1	1	1	0

to convert a given LFSR-based generator into a CA-based model is simple and can be applied to generators in a range of practical interest. The linearity of these cellular models can be advantageously used in the analysis and/or cryptanalysis of such keystream generators.

References

1. Bluetooth, Specifications of the Bluetooth system, Version 1.1 (February 2001), available at `http://www.bluetooth.com/`
2. Caballero-Gil, P., Fúster-Sabater, A.: A Wide Family of Nonlinear Filter Function with a Large Linear Span. Information Sciences 164(4), 197–207 (2004)
3. Cattell, K., Muzio, J.C.: Synthesis of One-Dimensional Linear Hybrid Cellular Automata. Computers-Aided Design 15(3), 325–335 (1996)
4. Coppersmith, D., Krawczyk, H., Mansour, Y.: The Shrinking Generator. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 22–39. Springer, Heidelberg (1994)
5. Fúster-Sabater, A.: Run Distribution in Nonlinear Binary Generators, Applied Mathematics Letters. Applied Mathematics Letters 17(12), 1427–1432 (2004)
6. Fúster-Sabater, A., Caballero-Gil, P.: Linear Automata in Cryptanalysis of Stream Ciphers. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 611–616. Springer, Heidelberg (2006)
7. Gollmann, D., Chambers, W.: Generators for Sequences with Near-Maximal Linear Equivalence. IEE Proceedings 135, 67–69 (1988)
8. Golomb, S.W.: Shift Register-Sequences. Aegean Park Press, Laguna Hill (1982)
9. Gong, G.: Theory and Applications of q-ary Interleaved Sequences. IEEE Trans. on Information Theory 41(2), 400–411 (1995)
10. GSM, Global Systems for Mobile Communications, available at `http://cryptome.org/gsm-a512.htm`
11. Kanso, A.: Clock-Controlled Shrinking Generators. In: Safavi-Naini, R., Seberry, J. (eds.) ACISP 2003. LNCS, vol. 2727, pp. 443–451. Springer, Heidelberg (2003)
12. Kari, J.: Theory of cellular automata: A survey. Theoretical Computer Science 334, 3–33 (2005)
13. Key, E.L.: An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators. IEEE Trans. Informat. Theory 22(6), 732–736 (1976)

14. Rueppel, R.A., Ciphers, S., Gustavus, i., Simmons, J.: Contemporary Cryptology, The Science of Information, pp. 65–134. IEEE Press, Los Alamitos (1992)
15. Serra, M., Slater, T., Muzio, J.C., Miller, D.M.: The Analysis of One-dimensional Linear Cellular Automata and Their Aliasing Properties. IEEE Trans. on Computer-Aided Design 9(7), 767–778 (1990)
16. Sun, X., Kontopidi, E., Serra, M., Muzio, J.C.: The Concatenation and Partitioning of Linear Finite State Machines. Int. J. Electronics 78, 809–839 (1995)

Effective Quantification of Gene Expression Levels in Microarray Images Using a Spot-Adaptive Compound Clustering-Enhancement-Segmentation Scheme

Antonis Daskalakis[1], Dionisis Cavouras[2], Panagiotis Bougioukos[1], Spiros Kostopoulos[1], Pantelis Georgiadis[1], Ioannis Kalatzis[2], George Kagadis[1] , and George Nikiforidis[1]

[1] Department of Medical Physics, School of Medicine, University of Patras, Rio, GR-26500 Greece
daskalakis@med.upatras.gr
[2] Medical Signal and Image Processing Lab, Department of Medical Instruments Technology, Technological Educational Institute of Athens, Ag. Spyridonos Street, Aigaleo, 122 10, Athens, Greece

Abstract. A spot-adaptive compound clustering-enhancement-segmentation (CES) scheme was developed for the quantification of gene expression levels in microarray images. The CES-scheme employed 1/griding, for locating spot-regions, 2/Fuzzy C-means clustering, for segmenting spots from background, 3/ background noise estimation and spot's center localization, 4/emphasizing of spot's outline by the CLAHE image enhancement technique, 5/segmentation by the SRG algorithm, using information from step 3, and 6/microarray spot intensity extraction. Extracted intensities by the CES-Scheme were compared against those obtained by the MAGIC TOOL's SRG. Kullback-Liebler metric's values for the CES-Scheme were on average double than MAGIC TOOL's, with differences ranging from 1.45bits to 2.77bits in 7 cDNA images. Coefficient-of-Variation results showed significantly higher reproducibility ($p<0.001$) for the CES-Scheme in quantifying gene expression levels. Processing times for 1024x1024 16-bit microarray images containing 6400 spots were 300 and 487 seconds for the CES-Scheme and MAGIC TOOL respectively.

Keywords: DNA, microarray image analysis, microarray griding. CLAHE, SRG.

1 Introduction

DNA microarrays are an experimental technology for exploring the genome. This technology provides to biomedical investigators a simple tool for monitoring the expression levels of thousands of genes, under the same experimental conditions, and thus an easy way to identify and quantify gene expression levels for all genes in an organism [1-3].

Microarrays are arrays of glass microscope slides, in which thousands of discrete DNA sequences are printed by a robotic arrayer, thus, forming circular spots of known diameter. To compare the relative abundance of each of these gene sequences

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 555 – 565, 2007.
© Springer-Verlag Berlin Heidelberg 2007

in two DNA samples, the two samples are 1/ labeled with red and green fluorescent dye, respectively, 2/ mixed, and 3/ competitively hybridized to the microarray slide. The end product of the comparative hybridization experiment is scanned, using lasers that excite each dye on the appropriate wavelength. The relative fluorescence intensity between each dye on each spot, which, in turn, represents the relative expression level of the corresponding gene in both samples, is recorded in 2 array images, one for each dye [4].

In order to extract those relative intensities [5, 6], a series of image analysis techniques have been employed namely griding, spot segmentation, and intensity extraction [7, 8]. Griding is the process of identifying and locating the coordinates of each cell containing the spot; the cell is a rectangular region containing the pixels of both the spot and its background. Segmentation refers to the classification of cell-pixels as either signal (spot's foreground) or surrounding area (spot's background). Spots' intensity extraction refers to the calculation of the fluorescence signal's mean intensity from the spot's foreground. Extracted mean intensities correspond to gene expression levels that, in turn, are translated into biological conclusions from molecular biologists, by employing data mining techniques.

A major factor that complicates the task of image analysis and data mining is that microarray experiments involve a number of error-prone steps (occurring during fabrication, target labeling, and hybridization), which induce noise on the resulting images [9, 10]. Microarray images are also corrupted by irregularities in the shape, size, and position of the spot [10, 11]. Improper treatment of noise may result to erroneous biological conclusions.

One of the most undesirable effects of noise is that it contributes to inaccurate spot segmentation (i.e. the boundaries of spots are incorrectly estimated), which leads to wrong estimation of the mean spots' intensities and reduces the reproducibility of the gene expression levels, derived from microarray images [12]. Although a variety of image pre-processing techniques have been suggested for correcting these sources of error [13], existing software tools, utilized for the analysis of microarray images, focus mainly on accurate spot localization and segmentation by various segmentation techniques [14-19]. Only few studies [20-24] have examined the impact of image preprocessing on cDNA image quality, however, without evaluating the effect of image enhancement on facilitating spot segmentation and, thus, on increasing the repeatability of the extracted gene expression levels.

In the present study, an efficient spot-adaptive compound clustering-enhancement-segmentation (CES) scheme is proposed for the quantification of gene expression levels in microarray images. The CES-scheme suitably incorporates the benefits of image enhancement into the spot segmentation procedure. Accordingly, the CES-scheme combines 1/a griding algorithm for locating individual spots 2/the Fuzzy C-means clustering algorithm [25], for automatically differentiating the spot's foreground and background, 3/a procedure for the assessment of local noise from the spot's background and for the determination of the spot's center, 4/the contrast limited adaptive histogram equalization (CLAHE) technique, for enhancing individual spot boundaries [13], 5/the seeded region growing (SRG) [26] segmentation technique, for segmenting microarray spots, employing information from step 3, and 6/a procedure for quantification of individual spot intensities. The proposed CES-scheme was implemented as part of a custom software that we have developed in our

lab [24]. The efficiency of the proposed CES-scheme was assessed employing both the Kullback-Liebler divergence metric [27], for goodness of segmentation, and the Coefficient of Variation metric, for assessing the reproducibility of extracted genes expression. Results were compared against those obtained by the SRG technique employed within the MAGIC TOOL's software [28], using a set of publicly available microarray cDNA images [29].

2 Material and Methods

Microarrays used in the current study comprised a public available dataset of 7 images, downloaded from the MicroArray Genome Imaging & Clustering Tool (MAGIC) website [29]. Each image of the dataset contained 6400 spots investigating the diauxic shift of *Saccharomyces cerevisiae.* In the particular dataset, the authors [30] have used a common reference messenger RNA pool (green, Cy-3) to control for biological variability [2, 31, 32]. Such a design provides an adequate degree of replication (7 images), required for the quantitative assessment of image segmentation and subsequent gene quantification.

2.1 CES-Scheme

The CES-Scheme was initialized by applying a griding procedure on the images. Ideally, spots are located at certain positions on the rectangular grid. By summing up the intensities across the pixels in each row and each column of the grid (line profiles), each spot center was represented by a peak-valley pattern, where peaks corresponded to spot centers and valleys to spot sites edges. Smoothing the line profiles by the Lowess filter [33], it ensured minimization of irregularities, introduced by the printing procedure, and, therefore, success of the griding procedure. Spot sites, in terms of width and height, were finally estimated from the peak-valley distance in each line profile. Thus, rectangular spot-containing regions (cell-regions) were formed.

Within each cell-region, the Fuzzy C-Means unsupervised algorithm [25] was employed to discriminate spots from surrounding background. The Fuzzy C-Means searched iteratively for cluster centers (centroids) that minimize the dissimilarity function (eq.1)

$$J(U, c_1, c_2, ..., c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2} \quad (1)$$

where c_i is the centroid of cluster i, d_{ij} is the Euclidean distance between *i-th* centroid (c_i) and *j-th* data point, U is the membership function matrix, and m is a weighting exponent (m=2). Subsequently, from the segmented spot and background, the location of the spot's center and local noise (standard deviation of background) respectively were assessed.

Following Fuzzy C-Means rough segmentation, spots in the individual cell-regions were enhanced using the CLAHE method [34]. CLAHE functioned adaptively on each cell-region aiming to maximize the contrast of each cell-spot relative to its background. The basic steps for its implementation were: a) division of the cell-region into a number of non-overlapping contextual regions of equal sizes (in our case regions were experimentally set to be approximately 40 pixels), b) computation of the histogram of each contextual region, c) enhancement of the contrast of each contextual region, by clipping its respective histogram under a certain threshold (t=0.001), d) redistribution of the histogram in such a way that its height did not exceed the clip limit, e) bilinear interpolation of the neighboring contextual regions and modification of the cells'-region gray-levels according to the Cumulative Distribution Function (CDF) (eq.2) of each contextual region.

$$f(n) = \frac{(N-1)}{M} \sum_{k=0}^{n} h(k) \tag{2}$$

where M and N are the numbers of pixels and gray-levels in the cell-region respectively and h(k) for k=0,1,…,N-1 is the histogram of each cell-region.

Contrast enhanced cell-regions were segmented using the SRG algorithm. SRG initially segmented each cell-region into regions of pixels starting from the spot's center, as determined by the Fuzzy C-Means segmentation. Pixel regions were iteratively augmented by assigning neighboring pixels that satisfied a homogeneity criterion: the neighboring pixels should be 1/of higher intensity than local noise, as it was calculated during the rough Fuzzy C-Means segmentation stage, and 2/of intensity close to the mean intensity of the so far seeded region. This iterative procedure of growing pixel regions within each cell-region continued until all pixels of the cell-region were assigned to either the spot or its background.

The spot's boundary, thus determined, was referred to the corresponding cell spot on the original image and the spot's intensity was evaluated. This was necessary, since intensities in the processed cell-spots were altered by the enhancement process.

2.2 Evaluation of Extracted Genes Expression Levels

Foreground (spot) and background intensity values for the common reference channel (green) were subsequently extracted and, considering all segmented spots for each image, two density distributions were produced, employing a non-parametric kernel density estimation method [35]. The distance between those two distributions was determined employing the Kullback-Liebler (K-L) measure of divergence, shown in (3):

$$K - L(P \parallel Q) = \sum_{i} P(i) \log \frac{P(i)}{Q(i)} \tag{3}$$

where P and Q are the spot and background density distributions respectively. Higher values of divergence correspond to more distant distributions and, consequently, to

more accurate segmentation, considering that intensities are evaluated on the original image alone.

Exploiting the benefits of the replicated common reference channel (Cy-3), provided through the design of the particular dataset, we quantitatively assessed the performance of the CES-Scheme in terms of extracted genes expression reproducibility using the Coefficient of Variation metric (*CV*) (eq.4).

$$CV = \frac{\sigma}{\mu} \tag{4}$$

where σ is the standard deviation and μ is the mean value for each spot evaluated for the all the replications (7 replications totally).

Extracted intensities employing the CES-Scheme were comparatively evaluated, in terms of both CV and *K-L* divergence, against the intensities obtained by a recently published commercial software (MAGIC TOOL [28]). For evaluation purposes, the same microarray images were introduced to both methods and the SRG segmentation option of the MAGIC TOOL was chosen, for fairness of comparison.

3 Results

Figure 1 shows the result of the successive griding procedure for two selected sub-blocks of the microarray images used in the present study.

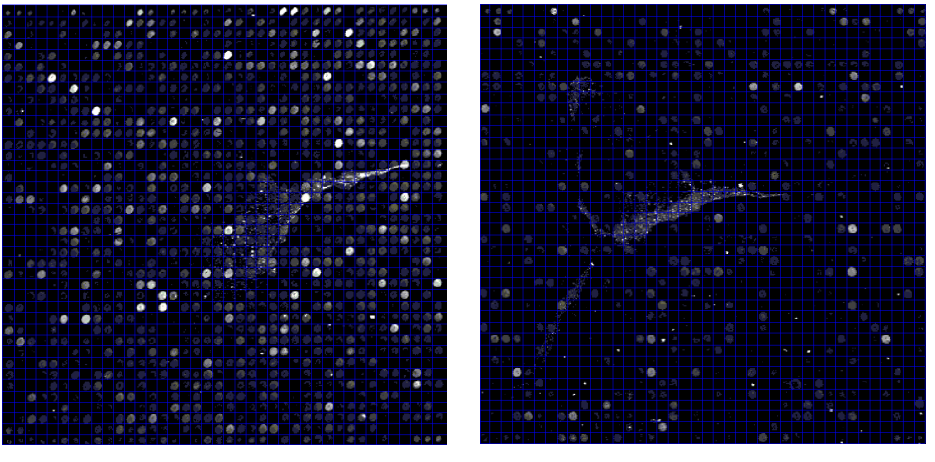


Fig. 1. Results of the griding procedure

Figure 2 shows the result of the Fuzzy C-Means (Fig. 2b), for rough estimation of spot and background, and the resulting segmented image (Fig. 2c), with spots' boundaries outlined, employing the proposed CES-Scheme.

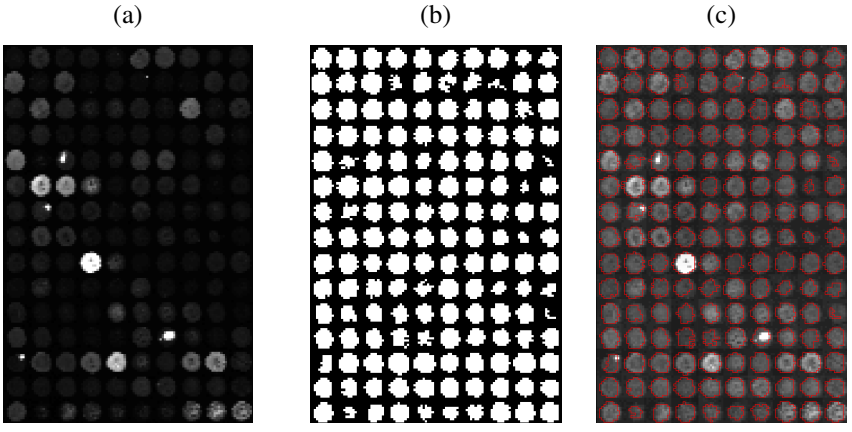


Fig. 2. a/Original image, b/Fuzzy C-means segmented image, and c/enhanced image, result of the CES-Scheme for microarray boundary determination

Table 1 presents the results of the Kullback-Liebler divergence between spot and background intensity distributions (of the common reference channel Cy-3) for the proposed methodology and the MAGIC TOOL respectively.

Table 1. Kullback-Liebler divergence metric values (in bits) between spot (signal) and background intensity values for the green (common reference sample) channel, for the seven evaluated cDNA images

Images	CES-Scheme	MAGIC TOOL
1302_OD370	10.620	4.807
1303_OD014	8.075	2.917
1309_OD690	10.620	7.284
1310_OD046	7.546	4.858
1311_OD080	6.404	4.405
1312_OD180	10.505	4.680
1313_OD370	8.253	5.127

Figure 3 depicts a randomly selected spot as appeared in 5 out of the 7 replicated images for the common reference channel. The middle row shows the result of the SRG algorithm on the latter mentioned spot according to the MAGIC TOOL software, and the bottom row presents the segmentation results based on the CES-Scheme.

Figure 4 illustrates the actual distribution of *CV* values of extracted gene expression levels from the set of seven 1024x1024 16-bit replicated images (Cy-3), each containing 6400 spots, as they were calculated by Magic Tool and the proposed CES-Scheme.

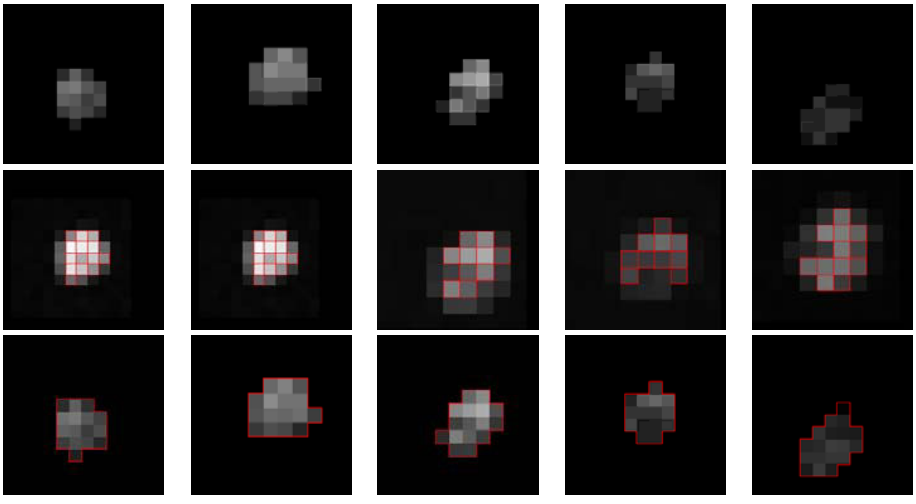


Fig. 3. Spot-containing cell region as appeared in 5 replica images and segmentation results according to the MAGIC TOOL's SRG (middle row) and the CES-Scheme (bottom row)

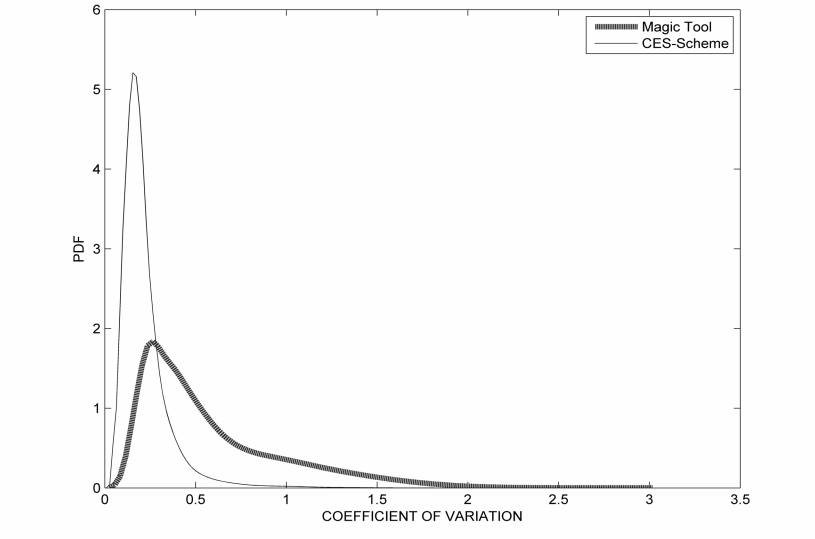


Fig. 4. Probability Density Functions (PDF's) of the Coefficient of Variation for all the spots as evaluated from the 7 replicated images of the common reference channel. Thick line corresponds to the results obtained using the MAGIC TOOL's SRG and thin line to the results obtained using the CES-Scheme.

4 Discussion

DNA microarrays represent a technological intersection between biology and computer science, which enables the gene expression analysis on a genome-wide scale. A

unique capability provided by the microarray technology is the collection of the necessary data for genomic analysis from just one experiment.

The major challenges of this approach appear in the image processing stage in which the spotted DNA sequences have to be extracted from the produced microarray images and quality measures have to be calculated. The accuracy of this stage has a straightforward impact on the accuracy and the effectiveness of the subsequent gene expression and identification analysis.

The task of microarray image processing is complicated, since the microarray experiments involve a number of error-prone steps, which induce noise on the produced images [9, 10]. Noise contributes to inaccurate spot segmentation (i.e. the boundaries of spots are incorrectly estimated) and, thus, indirectly it evokes the wrong estimation of the extracted genes expression levels [12].

Despite the variety of existing image pre-processing techniques that address noise effects [13], existing software tools focus mainly on spot recognition (spot localization) and extraction of genes' intensities employing various segmentation techniques [14-19]. Nevertheless, as a previous study has indicated [12], different segmentation methods, while accurate in simulated microarray images, lead to a different number of differentially expressed genes, when applied to identical real microarray images. Thus, a solution to the problem of accurate gene quantification might be provided by an intermediate pre-processing step.

Only few studies have examined the impact of image preprocessing techniques on cDNA image quality [20-23]. However, most of them were applied globally on the microarray images without taking into account the specific characteristics of individual spots [36]. Moreover, to the best of the authors' knowledge, there are no studies evaluating the effect of image enhancement in facilitating spot segmentation and, thus, in increasing the repeatability of the extracted gene expression levels.

In the present study, a compound Clustering-Enhancement-Segmentation (CES) scheme was developed to efficiently quantify genes expression levels in microarray images. The proposed compound CES-Scheme comprised 1/a griding algorithm for locating individual spot containing cell-regions, 2/the Fuzzy C-Means algorithm for automatically segmenting each cell-region into spot and background, 3/a procedure for assessing local noise and the spot's center, 4/the contrast limited adaptive histogram equalization (CLAHE) technique, for enhancing individual spot boundaries, 5/the seeded region growing (SRG) segmentation technique, for outlining the spot's boundary, and 6/a routine for quantifying the intensities of individual microarray spots on the original image.

Observing the original and the corresponding segmented images, as Figure 2 depicts, supports our initial hypothesis that an intermediate step of spot-based image enhancement improves the display of spots and emphasizes the depiction of spot edges even for those spots that are not well-defined (low intensity spots). Such an intermediate step resulted in more accurate and reproducible segmentation results, since it facilitated the spot's edge-detection. This conclusion is supported by the results depicted in Figure 3, where the same spot was segmented for 5 replicated images of the common reference channel using the MAGIC TOOL's SRG (middle row) and the proposed CES-Scheme (end row) .

To quantify the effectiveness of the proposed CES-Scheme, the information theoretic metric of the Kullback-Liebler divergence (Table 1) was employed and our

results were comparatively evaluated against those obtained using publicly available software (MAGIC TOOL). Results, according to the proposed scheme, were obtained by superimposing spot-outlines from the processed cell-region images on the original cell-spots. In this way, a higher Kullback-Liebler divergence, achieved by a particular method between the actual spots and its surrounding background, would eventual lead to better spot-boundary detection result. Table 1 confirms that the proposed scheme performed as anticipated, by increasing the divergence (K-L) between signal and background intensity distributions, as compared to corresponding distributions obtained using the MAGIC TOOL software.

The dataset used in the current study has been designed to control for biological variability (green channel was a common reference channel) [31]. Hence, an adequate degree of replication has been provided to quantitatively assess the reproducibility of the intensities extracted by the proposed CES-Scheme. Normally, replicate experiments have been used for reducing experimental variation [37]. Due to the replication provided, each spot should have the same intensity throughout the replicated experiments, and therefore the coefficient of variation between replicated experiments should be minimal (as close as possible to zero). Figure 4 shows the probability density functions (PDF) of the coefficient of variation for the common reference channel for all the images in the dataset using the CES-Scheme (thin line) and the MAGIC TOOL's SRG (thick line). CES-Scheme's PDF is narrow and sharp with a peak-value close to zero in contrast to MAGIC TOOL's PDF, which is more spread and far from zero. Distributions were found significantly different ($p<0.001$), employing Matlab's non-parametric statistical test (signtest).

Regarding processing time, the CES-Scheme took 300 seconds against MAGIC TOOL's 487 secs for the same 1024x1024, 16-bit cDNA image, containing 6400 microarray spots, and on the same computer.

Acknowledgments. This work was supported by a grant form the General Secretariat for Research and Technology, Ministry of Development of Greece (136/PENED03) to B.A.

References

1. Alizadeh, A., Eisen, M., Botstein, D., Brown, P.O., Staudt, L.M.: Probing lymphocyte biology by genomic-scale gene expression analysis. J Clin Immunol. 18, 373–379 (1998)
2. Churchill, G.A.: Fundamentals of experimental design for cdna microarrays. Nat Genet. 32, 490–495 (2002)
3. Taniguchi, M., Miura, K., Iwao, H., Yamanaka, S.: Quantitative assessment of DNA microarrays–comparison with northern blot analyses. Genomics. 71, 34–39 (2001)
4. Schena, M.: Microarray biochip technology, 1st edn. Eaton Publishing Company (2000)
5. Chen, Y., Dougherty, E., Bittner, M.: Ratio-based decisions and the quantitative analysis of cdna microarray images. Journal of Biomedical Optics 2, 364–374 (1997)
6. Schena, M.: Microarray analysis, 1st edn. New York (2002)
7. Jain, A.N., Tokuyasu, T.A., Snijders, A.M., Segraves, R., Albertson, D.G., Pinkel, D.: Fully automatic quantification of microarray image data. Genome Res. 12, 325–332 (2002)
8. Yang, Y.H., Buckley, M.J., Speed, T.P.: Analysis of cdna microarray images. Brief Bioinform. 2, 341–349 (2001)

9. Schuchhardt, J., Beule, D., Malik, A., Wolski, E., Eickhoff, H., Lehrach, H., Herzel, H.: Normalization strategies for cdna microarrays. Nucleic Acids Res. 28, 47 (2000)
10. Balagurunathan, Y., Wang, N., Dougherty, E.R., Nguyen, D., Chen, Y., Bittner, M.L., Trent, J., Carroll, R.: Noise factor analysis for cdna microarrays. J Biomed Opt. 9, 663–678 (2004)
11. Balagurunathan, Y., Dougherty, E.R., Chen, Y., Bittner, M.L., Trent, J.M.: Simulation of cdna microarrays via a parameterized random signal model. J Biomed Opt. 7, 507–523 (2002)
12. Ahmed, A.A., Vias, M., Iyer, N.G., Caldas, C., Brenton, J.D.: Microarray segmentation methods significantly influence data precision. Nucleic Acids Res. 32, e50 (2004)
13. Gonzalez, R.C., Woods, R.E.: Digital image processing, 1st edn (1992)
14. Axon Instruments. Genepix4000a user's guide (1999)
15. Steinfath, M., Wruck, W., Seidel, H., Lehrach, H., Radelof, U., O'Brien, J.: Automated image analysis for array hybridization experiments. Bioinformatics. 17, 634–641 (2001)
16. White, A.M., Daly, D.S., Willse, A.R., Protic, M., Chandler, D.P.: Automated microarray image analysis toolbox for matlab. Bioinformatics. 21, 3578–3579 (2005)
17. Zapala, M.A., Lockhart, D.J., Pankratz, D.G., Garcia, A.J., Barlow, C., Lockhart, D.J.: Software and methods for oligonucleotide and cdna array data analysis. Genome Biol. 3, SOFTWARE0001,0001-0001,0009 (2002)
18. QuantArray Analysis Software, O.s.M. Available: via the INTERNET. Accessed
19. Eisen, M.B.S.: (Accessed 06/12/2006) via the INTERNET, Available: http://rana.stanford.edu/software
20. Wang, X.H., Istepanian, R.S., Song, Y.H.: Microarray image enhancement by denoising using stationary wavelet transform. IEEE Trans Nanobioscience. 2, 184–189 (2003)
21. Lukac, R., Plataniotis, K.N., Smolka, B., Venetsanopoulos, A.N.: Cdna microarray image processing using fuzzy vector filtering framework. Journal of Fuzzy Sets and Systems: Special Issue on Fuzzy Sets and Systems in Bioinformatics (2005)
22. Mastriani, M., Giraldez, A.E.: Microarrays denoising via smoothing of coefficients in wavelet domain. International Journal of Biomedical Sciences. 1, 1306–1316 (2006)
23. Lukac, R., Smolka, B.: Application of the adaptive center-weighted vector median framework for the enhancement of cdna microarray. Int. J. Appl. Math. Comput. Sci., 13, 369–383 (2003)
24. Daskalakis, A., Cavouras, D., Bougioukos, P., Kostopoulos, S., Argyropoulos, C., Nikiforidis, G.C.: Improving microarray spots segmentation by k-means driven adaptive image restoration. In: Proceedings of the ITAB Ioannina, Greece (2006)
25. Jain, A.K.: Fundamentals of digital image processing. Prentice-Hall, Englewood Cliffs (1989)
26. Adams, R., Bischof, L.: Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence. 16, 641–647 (1994)
27. Kullback, S.: Information theory and statistics, 2nd edn. Dover Publications, Mineola (1968)
28. Heyer, L.J., Moskowitz, D.Z., Abele, J.A., Karnik, P., Choi, D., Campbell, A.M., Oldham, E.E., Akin, B.K.: Magic tool: Integrated microarray data analysis. Bioinformatics. 21, 2114–2115 (2005)
29. (Accessed 06/12/2006) Available: via the INTERNET. http://www.bio.davidson.edu/projects/MAGIC/MAGIC.html
30. DeRisi, J.L., Iyer, V.R., Brown, P.O.: Exploring the metabolic and genetic control of gene expression on a genomic scale. Science. 278, 680–686 (1997)

31. Sterrenburg, E., Turk, R., Boer, J.M., van Ommen, G.B., den Dunnen, J.T.: A common reference for cdna microarray hybridizations. Nucleic Acids Res. 30, e116 (2002)
32. Yang, Y.H., Speed, T.: Design issues for cdna microarray experiments. Nat Rev Genet. 3, 579–588 (2002)
33. Cleveland, W.S.: Robust locally weighted regression and smoothing scatterplots. Journal of the American Statistical Association, 829–836 (1979)
34. Pizer, S.M., Amburn, E.P.: Adaptive histogram equalization and its variations. Grpahics, and Image Processing 39, 355–368 (1987)
35. Bowman, A.W., Azzalini, A.: Applied smoothing techniques for data analysis. Oxford University Press, Oxford (1997)
36. Nykter, M., Aho, T., Ahdesmaki, M., Ruusuvuori, P., Lehmussola, A., Yli-Harja, O.: Simulation of microarray data with realistic characteristics. BMC Bioinformatics 7, 349 (2006)
37. Dudoit, S., Fridlyand, J., Speed, T.P.: Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association 97, 77 (2002)

Biomarker Selection, Employing an Iterative Peak Selection Method, and Prostate Spectra Characterization for Identifying Biomarkers Related to Prostate Cancer

Panagiotis Bougioukos[1], Dionisis Cavouras[2], Antonis Daskalakis[1], Ioannis Kalatzis[2], George Nikiforidis[1], and Anastasios Bezerianos[1].

[1] Department of Medical Physics, School of Medicine, University of Patras, Rio , GR-26500 Greece
[2] Medical Signal and Image Processing Lab, Department of Medical Instruments Technology, Technological Educational Institute of Athens, Ag. Spyridonos Street, Aigaleo, 122 10, Athens, Greece

Abstract. A proteomic analysis system (PAS) for prostate Mass Spectrometry (MS) spectra is proposed for differentiating normal from abnormal and benign from malignant cases and for identifying biomarkers related to prostate cancer. PAS comprised two stages, 1/a pre-processing stage, consisting of MS-spectrum smoothing, normalization, iterative peak selection, and peak alignment, and 2/a classification stage, comprising a 2-level hierarchical tree structure, employing the PNN and SVM classifiers at the 1st (normal-abnormal) and 2nd (benign-malignant) classification levels respectively. PAS first applied local thresholding, for determining the MS-spectrum noise level, and second an iterative global threshold estimation algorithm, for selecting peaks at different intensity ranges. Two optimum sub-sets of these peaks, one at each global threshold, were used to optimally design the hierarchical classification scheme and, thus, indicate the best m/z values. The information rich biomarkers 1160.8, 2082.2, 3595.9, 4275.3, 5817.3, 7653.2, that have been associated with the prostate gland, are proposed for further investigation.

Keywords: Mass-Spectrometry; Biomarker Selection; Classification.

1 Introduction

Early detection of cancer is a critical issue for improving patient survival rates. Regarding prostate cancer, it is the second leading cause of cancer deaths in United States and Canada [1]. In daily clinical practice, the most widely used method for prostate cancer detection is the measurement of the prostate specific antigen (PSA). The PSA diagnostic test exhibits high sensitivity. However, its low specificity confines its use as an early detection biomarker. This calls for the discovery of novel biomarkers that will result in higher specificity, thus, aiding in the decrease of prostate cancer mortality.

With recent advances in mass spectrometry techniques (MS), it is now possible to investigate proteins over a wide range of molecular weights in small biological specimens, such as serum [2]. However, MS proteomic profiles have several imperfections, which

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 566 – 574, 2007.
© Springer-Verlag Berlin Heidelberg 2007

may complicate their interpretation [3]. Thus, data preprocessing steps, such as smoothing, normalization, peak detection, and peak alignment may improve the performance of proteomic analysis and assist in biomarker discovery [4].

Regarding prostate cancer, previous studies [5-10] have implemented various preprocessing algorithms and have applied different pattern recognition techniques for proposing biomarkers, which however differ. Classification accuracies that have been achieved in some of those studies ranged between 73% and 100% [5, 9-11], however, using own data, while in two studies [6, 7], use has been made of the same with us dataset, achieving 97% and 92% classification accuracies. The latter have been attained by the inclusion of less than 1000 m/z values, which, however, have been reported to be of non-significance due to distortion [10].

In the present study, a proteomic analysis system (PAS) for prostate Mass Spectrometry (MS) spectra is proposed for differentiating normal from abnormal and benign from malignant cases and for identifying biomarkers related to prostate cancer. PAS comprised two stages, 1/a pre-processing stage, consisting of MS-spectra smoothing, normalization, iterative peak selection, and peak alignment, and 2/a classification stage, comprising a 2-level hierarchical tree structure employing the PNN and SVM classifiers at the 1st (normal-abnormal) and 2nd (benign-malignant) classification levels respectively. PAS first applied local thresholding, for determining the MS-spectrum noise level, and second an iterative global threshold estimation algorithm, for selecting peaks at different intensity ranges. Two optimum sub-sets of these peaks, one at each global threshold, were used to optimally design the hierarchical classification scheme. Among those peaks, prostate related biomarkers were identified by reference to published literature and public databases.

2 Materials and Methods

Mass spectrometry prostate cancer spectra were obtained from the National Cancer Institute Clinical Proteomics Database [12]. MS-spectra were produced utilizing the H4 protein chip and a Ciphergen PBS1 SELDI-TOF mass spectrometer. The chip was prepared by hand and MS-spectra were exported with baseline subtracted. The dataset comprised 63 spectra, with no evidence of disease (PSA<1), 69 prostate cancer spectra (PSA>4), and 69 benign spectra (PSA>4).

2.1 PAS Pre-processing Stage

Signal noise residues were diminished applying the Lowess smoothing technique [13] (see Fig. 1), a function provided by Matlab's bioinformatics toolbox. After smoothing, all spectra were normalized to their total ion current and spectra were scaled to have an overall maximum intensity of 100.

Following smoothing, all MS-spectra were further processed utilizing an iterative peak selection method. Accordingly, the steps employed were:

a/ For each spectrum the noise level was estimated by dividing the spectrum into four equal parts. In this way, local thresholds were determined and were further used to determine intensities (above local thresholds) and noise (below local thresholds) (see Fig. 2).

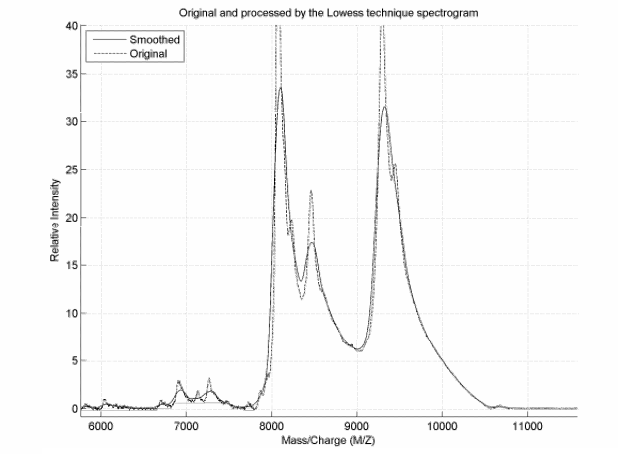


Fig. 1. Original and processed by the Lowess technique spectrogram

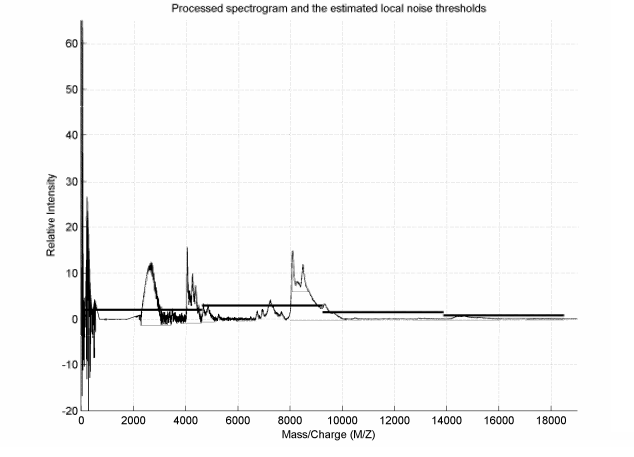


Fig. 2. Local noise threshold estimation for each mass spectrum

Local thresholds were determined by first computing at each quartile the histogram of intensities and then by locating the histogram's maximum value (see Fig 3).

b/ Intensity values above local thresholds were searched for peaks, by applying a differentiation method between successive intensity data points.

c/ Selected peaks were used to form a histogram of intensities, and its maximum value constituted the global threshold, above which all peaks were considered as most informative and were stored for further processing.

d/ Peaks determined in step **c** were removed from the spectrum and step **c** was repeated until all peaks of the entire spectrum were removed.

Steps **a-d** were repeated for all the MS-spectra of each class. In this way, peaks were grouped according to their global threshold level.

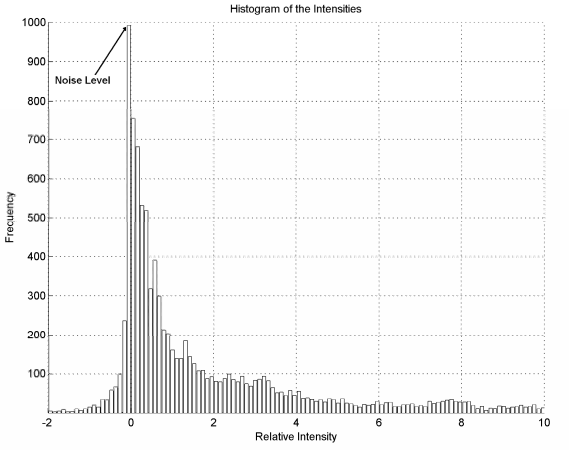


Fig. 3. Histogram of the intensities across m/z values. The maximum of the histogram depicts the average mass spectrum noise intensity level.

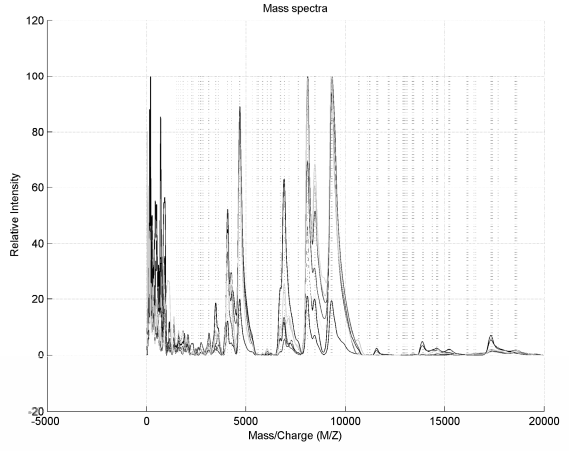


Fig. 4. Seven control cases mass spectra. The dotted vertical lines represent the corresponding m/z values of the peaks.

At each global threshold level, each MS-spectrum was represented by a varying number of peaks due to chemical and electronic noise [14]. To alleviate this, a peak alignment process was developed, that aligned peaks appearing concurrently in 5% of the available spectra but sustaining a small shift along the x-axis, 0.08 % and ignoring the rest (see Fig 4).

After peak alignment at each global threshold, peaks were further used to optimally design PNN and SVM classifiers (see section 2.2).

2.2 PAS Classification Stage

A two-level hierarchical tree structure was constructed for differentiating normal from abnormal (1st level) and benign from malignant (2nd level) cases employing:

1/ At the 1st level a Probabilistic Neural Network (PNN) classifier with discriminant function given by [15] (eq. 1) :

$$d_i(x) = \frac{1}{(2\pi)^{d/2}\sigma^d}\frac{1}{N}\sum_{k=1}^{N}\exp\left[-\frac{(x-x_{ik})^T(x-x_{ik})}{2\sigma^2}\right] \quad (1)$$

where σ is the spread of the Gaussian activation function that was experimentally determined and was set to *0.5*, N is the number of pattern vectors, d is the dimensionality of pattern vectors and x_{ik} is the k^{th} pattern vector of class i.

2/ At the 2nd level, a Support Vector Machines (SVM) classifier was employed with discriminant function [16] given by (eq. 2).

$$g(x)=sign\left(\sum_{i=1}^{N}a_i y_i K(x,x_i)+b\right) \quad (2)$$

where x_i is the training data belonging to class i, y_i is the class label {-1,+1}, N is the number of pattern vectors, α_i, b are weight coefficients and K the transformation or kernel function [16]. The Kernel function utilized was the radial basis function (RBF) (eq.3).

The optimization problem of calculating parameters α_i, was solved by using the function quadprog provided by the MATLAB's optimization toolbox.

$$K_{RBF}(x,x_i)=\exp\left(\frac{-|x-x_i|^2}{2\sigma^2}\right) \quad (3)$$

At each global threshold level, feature selection for both classifiers was performed in order to retain only the feature subset with the highest discriminatory information. For this purpose, the sequential backward feature selection algorithm was employed [17]. Classifiers' evaluation was performed by the standard procedure of the leave-one-out method (LOO) [17]. The classifiers were designed employing features from all but one MS-spectrum, which was then classified. The process was repeated, each time leaving-out a different MS-spectrum, until all data were processed. In this way, classifiers were evaluated by spectra not involved in their design. The above method has been widely used in many studies, with limited number of data [11, 18, 19]. Among the set of best features employed in the optimal design of the hierarchical tree structure, prostate related biomarkers were identified by reference to published literature and public databases [20].

3 Results

In distinguishing spectra with no evidence of disease (PSA<1) from spectra of benign and malignant nature (PSA≥4) PAS scored 94.5% and 93% overall accuracy (see Tables 1 and 2) at the 1st and 2nd global threshold levels. The m/z values corresponding to the best features obtained at the 2 classification levels of the hierarchical tree structure and for the two aforementioned global thresholds are illustrated in Table 3. Table 4 illustrates a subset of the m/z values, shown in Table 3, corresponding to verified biomarkers, which were identified by reference to published literature and public databases [20]. These m/z values are proposed as useful biomarkers.

Table 1. PAS classification results at the first global threshold level, for spectra with no evidence of disease and spectra of benign and malignant nature

	Normal	Benign	Malignant	Accuracy (%)
Normal	62	1	0	98.4
Benign	1	64	4	92.8
Malignant	1	4	64	92.8
Overall accuracy				94.5

Table 2. PAS classification results at the second global threshold level, for spectra with no evidence of disease and spectra of benign and malignant nature

	Normal	Benign	Malignant	Accuracy (%)
Normal	62	1	0	98.4
Benign	4	62	3	89.9
Malignant	3	3	63	91.3
Overall accuracy				93

Table 3. m/z values of the best feature vectors

Global Threshold levels	Feature vectors (biomarkers)
First	{ 1160.8, 1838.2, 3595.9, 4074.3, 4257.1, 4275.3, 6918.2, 8449.1, 8470.2, 9317.5, 9767.5, 14609.9, 4087.9, }
Second	{ 1060.5, 1298.8, 1468.5, 1755.6, 2082.2, 2213.6, 2736.1, 4503.7, 4672, 5326.0, 5817.3, 10673.1, 3607.1, 6939.1, 7653.2}

Table 4. Verified biomarkers selected from the two best feature vectors

Global Threshold levels	Feature vectors (biomarkers)
First	{ 1160.8, 3595.9, 4275.3 }
Second	{ 2082.2, 4672, 5817.3, 7653.2}

4 Discussion

The goal of this study was to develop a proteomic analysis system for prostate MS-spectra characterization into normal, benign, and malignant and to propose biomarkers related to prostate cancer. The proposed PAS-system achieved 94.5% and 93% at the first and the second global threshold levels respectively, in discriminating individuals with no evidence of disease (PSA<1) from those with prostate disease (benign or malignant, PSA≥4). Employing the proposed iterative peak selection method and by investigating the accuracies attained, it was plausible to conclude that peaks realizing high intensity values are not necessarily the most significant in distinguishing normal from abnormal prostate cases. This iterative peak selection method facilitates in reducing the number of potential biomarkers as well as in assisting researchers by focusing on specific intensity levels for biomarker discovery. Specifically, the highest discriminating power was achieved using the biomarkers that were revealed at specific global threshold levels.

Seeking for proteins that might be related to prostate cancer, it was found, by searching in the ExPASy database [20], that the m/z value of 2082.2 was very close to the Nociceptin protein (2081.39 Daltons), which has been implicated in the stimulation of prostate cell growth [21], among other neuropeptides. Additionally, the 4672 m/z value is very close to the BAX protein, cytoplasmic isoform gamma that has a molecular weight of 4678.22 (Daltons). This protein belongs to the BCL-2 gene family; however, as it has been reported in [22], elevated levels of BCL-2 might aid in the progression of the prostate cancer. Also, another m/z value (biomarker) was the 5817.3, which is very close to the 5818.62 (Granulin precursor). This precursor is a prostate cancer cell-derived growth factor and its expression has been found to be elevated in high-grade prostatic intraepithelial neoplasia and prostatic adenocarcinoma [23]. Additionally, the biomarker 7653.2 has m/z value close to 7654.74, which is the Insulin-like Growth factor I (IGF-I). This protein is a mitogen for prostate epithelial cells and it has been reported [24] to have a strong positive association between elevated levels of IGF-I and prostate cancer risk. The 1160.8 m/z value is close to 1169.32 biological marker, named prostasin precursor, which is present in many tissues and has the highest level in prostate gland [25]. One of the seminal plasma proteins realizing 3590.98 m/z value is very close to the 3595.9 m/z value, which was identified to be seminal basic protein (BSP), a proteolytic product of semenogelin 1 [26]. The 4275.3 has m/z value close to 4272.2, which is the Neuropeptide Y precursor and its activation might be related in cell proliferation in certain stages of prostate cancer [27]. Neuropeptide Y precursor (NPY) (3474.91 daltons) is close to 3474 m/z value and it has been reported [28] that NPY may directly regulate prostate cancer growth via Y1-R gene expression. This finding suggests that NPY related mechanism might play a relevant role in the progression of prostate cancer at both androgen dependent and independent stages. The present study revealed other m/z values as being significant in distinguishing prostate cancer dataset, as illustrated in Tables 3. These values might constitute information rich biomarkers that have not been yet identified or related to prostate cancer.

Acknowledgments. This work was supported by a grant form the General Secretariat for Research and Technology, Ministry of Development of Greece (013/PENED03) to A.B.

References

1. McDavid, K., Lee, J., Fulton, J.P., Tonita, J., Thompson, T.D.: Prostate cancer incidence and mortality rates and trends in the united states and canada. Public Health Rep. 119, 174–186 (2004)
2. Srinivas, P.R., Srivastava, S., Hanash, S., Wright Jr, G.L.: Proteomics in early detection of cancer. Clin Chem. 47, 1901–1911 (2001)
3. Hilario, M., Kalousis, A., Pellegrini, C., Muller, M.: Processing and classification of protein mass spectra. Mass Spectrom Rev. 25, 409–449 (2006)
4. Malyarenko, D.I., Cooke, W.E., Adam, B.L., Malik, G., Chen, H., Tracy, E.R., Trosset, M.W., Sasinowski, M., Semmes, O.J., Manos, D.M.: Enhancement of sensitivity and resolution of surface-enhanced laser desorption/ionization time-of-flight mass spectrometric records for serum peptides using time-series analysis techniques. Clin Chem. 51, 65–74 (2005)
5. Jong, K., Marchiori, E., Sebag, M., van der Vaart, A.: Feature selection in proteomic pattern data with support vector machines. In: Proceedings of the (2004) 41
6. Lilien, R.H., Farid, H., Donald, B.R.: Probabilistic disease classification of expression-dependent proteomic data from mass spectrometry of human serum. J Comput Biol. 10, 925–946 (2003)
7. Petricoin, E.F., Ornstein 3rd, D.K., Paweletz, C.P., Ardekani, A., Hackett, P.S., Hitt, B.A., Velassco, A., Trucco, C., Wiegand, L., Wood, K., Simone, C.B., Levine, P.J., Linehan, W.M., Emmert-Buck, M.R., Steinberg, S.M., Kohn, E.C., Liotta, L.A.: Serum proteomic patterns for detection of prostate cancer. J Natl Cancer Inst. 94, 1576–1578 (2002)
8. Qu, Y., Adam, B.L., Thornquist, M., Potter, J.D., Thompson, M.L., Yasui, Y., Davis, J., Schellhammer, P.F., Cazares, L., Clements, M., Wright Jr, G.L., Feng, Z.: Data reduction using a discrete wavelet transform in discriminant analysis of very high dimensionality data. Biometrics 59, 143–151 (2003)
9. Qu, Y., Adam, B.L., Yasui, Y., Ward, M.D., Cazares, L.H., Schellhammer, P.F., Feng, Z., Semmes, O.J., Wright Jr, G.L.: Boosted decision tree analysis of surface-enhanced laser desorption/ionization mass spectral serum profiles discriminates prostate cancer from non-cancer patients. Clin Chem. 48, 1835–1843 (2002)
10. Yasui, Y., Pepe, M., Thompson, M.L., Adam, B.L., Wright, G.L., Jr., Qu, Y., Potter, J.D., Winget, M., Thornquist, M., Feng, Z.: A data-analytic strategy for protein biomarker discovery: Profiling of high-dimensional proteomic data for cancer detection. Biostatistics 4, 449–463 (2003)
11. Adam, B.L., Qu, Y., Davis, J.W., Ward, M.D., Clements, M.A., Cazares, L.H., Semmes, O.J., Schellhammer, P.F., Yasui, Y., Feng, Z., Wright Jr, G.L.: Serum protein fingerprinting coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men. Cancer Res. 62, 3609–3614 (2002)
12. Institute, N.C. (Accessed 24/11/2006) via the INTERNET Available: http://home.ccr.cancer.gov/ncifdaproteomics/ppatterns.asp
13. Cleveland, W.S.: Robust locally weighted regression and smoothing scatterplots. J Amer Statist Assoc. 74, 829–836 (1979)

14. Baggerly, K.A., Morris, J.S., Coombes, K.R.: Reproducibility of seldi-tof protein patterns in serum: Comparing datasets from different experiments. Bioinformatics 20, 777–785 (2004)
15. Specht, D.F.: Probabilistic neural networks. Neural Networks 3, 109–118 (1990)
16. Christanini, N., Taylor, J.S.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000)
17. Theodorides, S., Koutroumbas, K.: Pattern recognition, 2nd edn. Academic Press, London (2003)
18. Ressom, H.W., Varghese, R.S., Abdel-Hamid, M., Eissa, S.A., Saha, D., Goldman, L., Petricoin, E.F., Conrads, T.P., Veenstra, T.D., Loffredo, C.A., Goldman, R.: Analysis of mass spectral serum profiles for biomarker selection. Bioinformatics. 21, 4039–4045 (2005)
19. Wang, X., Zhu, W., Pradhan, K., Ji, C., Ma, Y., Semmes, O.J., Glimm, J., Mitchell, J.: Feature extraction in the analysis of proteomic mass spectra. Proteomics 6, 2095–2100 (2006)
20. ExPASy. (Accessed 05/12/2006) Available via the INTERNET: http://au.expasy.org/tools/
21. Swanson, T.A., Kim, S.I., Myers, M., Pabon, A., Philibert, K.D., Wang, M., Glucksman, M.J.: The role of neuropeptide processing enzymes in endocrine (prostate) cancer: Ec 3.4.24.15 (ep24.15). Protein Pept Lett. 11, 471–478 (2004)
22. Hering, F.L., Lipay, M.V., Lipay, M.A., Rodrigues, P.R., Nesralah, L.J., Srougi, M.: Comparison of positivity frequency of bcl-2 expression in prostate adenocarcinoma with low and high gleason score. Sao Paulo Med J. 119, 138–141 (2001)
23. Pan, C.X., Kinch, M.S., Kiener, P.A., Langermann, S., Serrero, G., Sun, L., Corvera, J., Sweeney, C.J., Li, L., Zhang, S., Baldridge, L.A., Jones, T.D., Koch, M.O., Ulbright, T.M., Eble, J.N., Cheng, L.: Pc cell-derived growth factor expression in prostatic intraepithelial neoplasia and prostatic adenocarcinoma. Clin Cancer Res. 10, 1333–1337 (2004)
24. Chan, J.M., Stampfer, M.J., Giovannucci, E., Gann, P.H., Ma, J., Wilkinson, P., Hennekens, C.H., Pollak, M.: Plasma insulin-like growth factor-i and prostate cancer risk: A prospective study. Science 279, 563–566 (1998)
25. Yu, J.X., Chao, L., Chao, J.: Molecular cloning, tissue-specific expression, and cellular localization of human prostasin mrna. J Biol Chem. 270, 13483–13489 (1995)
26. Adam, B.L., Vlahou, A., Semmes, O.J., Wright Jr, G.L.: Proteomic approaches to biomarker discovery in prostate and bladder cancers. Proteomics 1, 1264–1270 (2001)
27. Magni, P., Motta, M.: Expression of neuropeptide y receptors in human prostate cancer cells. Ann Oncol. 12(2), S27–S29 (2001)
28. Ruscica, M., Dozio, E., Boghossian, S., Bovo, G., Martos Riano, V., Motta, M., Magni, P.: Activation of the y1 receptor by neuropeptide y regulates the growth of prostate cancer cells. Endocrinology 147, 1466–1473 (2006)

Classic Cryptanalysis Applied to Exons and Introns Prediction

Manuel Aguilar R., Héctor Fraire H., Laura Cruz R.,
Juan J. González B., Guadalupe Castilla V., and Claudia G. Gómez S.

Instituto Tecnológico de Ciudad Madero
manuel_itcm@yahoo.es,hfraire@prodigy.net.mx,
yolandaa@udc.es,lcruz@itcm.edu.mx,
{jgvaldez50,cggs71,jjgonzalezbarbosa}@hotmail.com

Abstract. Prediction of exonic and intronic regions is an important problem of bioinformatics, which has been solved with a set of medium accuracy coding measures. In this work, we propose a new methodology for the prediction of exons and introns based on a cryptanalysis method of Kasiski, using variants of three classical coding measures: codon usage, amino acid usage, and codon preference. We validated our approach testing a set of 178 sequence of different length, improving the prediction of exons level reported by Fickett. Additionally we introduce the first results of introns prediction with an accuracy level of 83.4%.

1 Introduction

DNA is the hereditary material located in the cell nucleus, the whole collection of DNA for an organism is called genome. DNA has a double helix shape composed by two intertwined chains made up of building blocks called nucleotides (also named bases). The nucleotides are represented by symbols of the alphabet $\Sigma = \{A, C, G, T\}$. The used symbols corresponds with Adenine, Cytosine, Guanine and Thymine respectively. There is a "lock-and-key" fit between the bases on opposite strands. Adenine pairs only with thymine and guanine pairs only with cytosine. A genome is composed by molecules of DNA extremely large organized on chromosomes.

The genes are DNA regions, most of them with the biological role to carry information specifying the chemical composition of proteins (gene exonic regions) and the regulatory signals that will govern the protein production by the cell (gene intronic regions) [1]. A gene becomes to be functional, it can synthesize a protein, when its DNA sequence passes through two global processes: transcription and translation. On the transcription process, a strand of the gene is mapped to a RNA sequence. In the DNA sequence, one of the intertwined chains is selected and each Thymine symbol is replaced by the base Uracil symbol (U). An example of the transcription process is given in Fig. 1.

After that a polyadenine is added to the end of sequence, the introns are cut and its exons are incorporated in the messenger RNA (mARN). Fig. 2 shows this process.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 575–584, 2007.
© Springer-Verlag Berlin Heidelberg 2007

DNA Chain 1	CTGCCATTGTCAGACATGTATACCCCGTACG
DNA Chain 2	GACGGTAACAGTCTGTACATATGGGGCATGC
RNA Chain	CUGCCAUUGUCAGACAUGUAUACCCCGUACG

Fig. 1. Transcription of a DNA strand

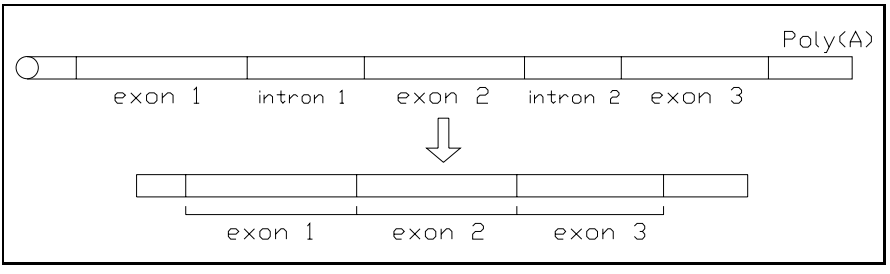


Fig. 2. Alternative Splicing

The translation is the process of producing a chain of amino acids based on a sequence of nucleotides in the mRNA. The nucleotide sequence of an mRNA molecule is read in groups of three successive bases called codons, the next example shows this process:

AUU CCG UAG GUA AAU UUG
Codon Codon Codon Codon Codon Codon

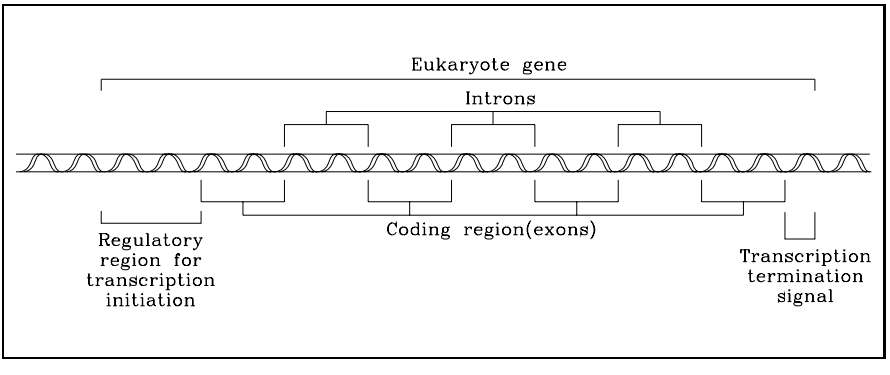


Fig. 3. Generalized gene structure in eukaryotes

The set of correspondences between the 64 possible codons in DNA and the 20 possible amino acids is named genetic code. Find a gene in a whole genome is a very hard problem, due this is necessary to characterize the gene in protein coding regions (exons) and protein non coding regions (introns)like is shown in Fig. 3. A formal definition of the prediction of exons and introns problem in a DNA sequence is:

Definition 1. *Determinate a function that maps a given DNA sequence C, with the probability LP of that the sequence to be protein coding region (exon) or not (intron). This kind of functions are named coding measures.*

There are several coding measures, the characteristics of these statistics measures are involved in the synthesis process of proteins. The usage of codons, amino acids, the prototype of a codon in a gene, are some coding measures commonly used. A coding measure has two models, the coding model of DNA and the non coding model of DNA. The first uses a distribution of a codon usage table, based on coding regions of knowledge genes. The non coding regions do not require a knowledge function in the protein synthesis. The non coding model uses a random distribution due to the non coding regions have a higher randomness grade than coding regions.

Homo sapiens genome has 3,000,000 kb size, where introns comprise 90%. In the actual paradigm the introns do not have an active participation in the synthesis process. But the bioinformatics is in constant growth and in the future the introns could have an important role in synthesis process. Our proposal combines coding measures biased by codon with the cryptanalysis Kasiski method. The approach includes a non random distribution for the non coding model of DNA in the measures, to improve the characterization of the introns.

2 Related Works

In this work we use elements of the classic cryptanalysis and coding measures. Respect to the classic cryptanalysis applied to the analysis of DNA sequence, Santos propose an approach based in the classic cryptanalysis without delimiting a particular problem of bioinformatics, and cannot conclude in a objective way [2]. On another hand we have the coding measure works. Staden creates the first measure named codon usage, measuring the occurrence of codons in regions of a genome, and comparing these frequencies with frequencies of coding regions priori measured [3]. McCaldon introduces amino acid usage, calculating the probabilities of occurrence of distinct oligopeptides in existing proteins, translating a codons sequence to a amino acids sequence[4]. Fickett computes the same measure based on bias amino acid computing the observed frequency of a single amino acid in existing proteins [5]. Gribskov introduces a measure called codon preference to avoid the uneven usage of synonymous codons in coding regions [6].In [5] the best measure is hexamer usage for predicting exons. In this work, we use three coding measures: codon usage, amino acid usage and codon preference because all of them use solely codons.

In Table 1, we can see the most important characteristics of the coding measures. Commonly the coding measures use a distribution of codon usage table(DCUT) for coding the DNA model, and a random distribution(RD) for the non coding model. In contrast we propose use a distribution generated with a classic cryptanalysis method. Like the results for prediction of exons are published but the results of prediction of introns are not, in this paper, we are publishing results of prediction for both: exons and introns [5].

Table 1. Characteristics of the coding measures

Coding Measure	Coding Model of DNA	Results	Non Coding Model of DNA	Results
HU	DCUT	[5]	RD	×
CU	DCUT	[5]	RD	×
AAU	DCUT	[5]	RD	×
CP	DCUT	[5]	RD	×
Our Proposal	DCUT	√	Cryptanalysis	√

3 Adaptation of Kasiski Method (KM)

The KM can be used to identify the length of the keyword used to create a ciphertext message encrypted by Vigenere cipher. The main idea of this method is the observation that any two identical substrings of the plaintext will encrypt to identical ciphertext substrings, when they are both aligned equally relative to the keyword boundaries. In this case the distance between the starting position of each substring is a multiple of the keyword length. In converse, if we identify multiple substrings of the ciphertext that are identical and of a sufficient length (say three or more letters long) it is highly probable that they are encryptions of identical plaintext. Thus, the length of the keyword must divide the greatest common divisor of the differences in starting positions of each of the ciphertext substrings [7]. The KM can be resumed in the following steps:

- Step 1. Analyze the repeated triplets in the ciphertext, keeping their starting position.
- Step 2. Calculate the difference between starting position of the repeated triplets of the ciphertext.
- Step 3.Determinate the length of the ciphertext's key, calculating the greatest common divisor of the obtained differences.

Table 2. Output of step 1 of KM applied to a ciphertext

Triplet	Occurrence	Positions
T_1	O_1	$P_1, ...P_i; i \geq 1$
..	..	..
T_N	O_N	$P_1, ...P_i; i \geq 1$

KM can compute the length of a Vigenere's cipher, in the first step of KM is possible to make a distribution of overlapped trigrams with an alphabet of N letters, it can be seen in Table 2. Also, the first step is used to make a distribution of overlapped trigrams (see Table 3) of a DNA sequence Cr, such as $C \subset Cr$ to use it instead of the random distribution of the coding measure when the coding potential of the DNA sequence C be computed.

Table 3. Output of KM applied to DNA sequence

Codon	Occurrence	Amino acid
C_1	O_{C_1}	A_{C_1}
..	..	..
C_{64}	$O_{C_{64}}$	$A_{C_{64}}$
	$TotalOcurrence$	

4 The Variants of Coding Measures Based on KM

Given two codons sequences C and Cr, such as, $C \subset Cr$, and a coding measure, we propose to replace the random model of coding DNA by the probability to find C. To validate this approach three variants were designed to discriminate if C is a sequence that codes a protein or just a sequence into Cr.

4.1 Variant of Codon Usage Based on KM(VCUKM)

VCUKM, similarly to [4], consider that the usage of codons in a region C of a genome for a species, with typical frequencies respect to a region of any gene of this species, could be a candidate to be a coding region for a protein in Cr. The probability $F(c)$ of codon c in coding region for the considered species, is given in a usage codon table. For a sequence C of codons, a frame is generated reading codons from a start position. Three different frames can be formed starting the reading process in the first, second or third positions. The coding protein probability of the codons sequence $C = c_1c_2c_3...c_m$, in frame i, is given by (1)(where c_j^i is the j codon in frame i). The probability $F^{'}(c)$ to appear the codon c into Cr is given by (2)(where the codon occurrences O_c and $TotalOccurrence$ are generated with the Kasiski method) . The probability to find a codon sequence C in frame i into Cr is computed as in (3).

$$P^i(C) = \prod_{j=1}^{m} F(c_j^i); 1 \leq i \leq 3 \tag{1}$$

$$F_{Cr}^{'}(c) = O_c/TotalOccurrence \tag{2}$$

$$P_{Cr}^i(C) = \prod_{j=1}^{m} F'(c_j^i); 1 \leq i \leq 3 \tag{3}$$

In (4), the protein coding potential of C in frame i is defined.

$$LP_{Cr}^i(C) = \log \frac{P^i(C)}{P_{Cr}^i(C)} \tag{4}$$

The protein coding potential is used to determine if C corresponds to an exonic or intronic region. If $\max LP_{Cr}^i(C) > 0$, then C is in an exonic region otherwise is in a intronic region.

4.2 Variant of Amino Acid Usage Based on KM(VAAUKM)

Similarly to amino acid usage [5], VAAUKM is a coding measure that is computed by amino acid bias based on the observed frequencies of a single amino acid in Cr. The probability of that the codon c, to be translated to an oligopeptide into Cr, $FA'_{Cr}(c)$, is the summing up of the probabilities of all the synonymous codons to c, that appear into Cr, as described in (5).

$$FA'_{Cr}(c) = \sum_{\forall c' \ synonymous of \ c} FA'_{Cr}(c') \tag{5}$$

The probability $PA^i_{Cr}(C)$ to find the sequence C of frame i, into Cr is computed as in (6).

The protein coding potential of C in frame i, $LPA^i_{Cr}(C)$, is computed in a similar way of (4).

$$PA^i_{Cr}(C) = \prod_{j=1}^{m} FA'_{Cr}(c^i_j); 1 \leq i \leq 3 \tag{6}$$

Also, the same conditions considered to verify if C corresponds with an exonic or intronic region are used.

4.3 Variant of Codon Preference Based on KM(VCPKM)

Similarly to codon preference [6], VCPKM is a coding statistic to measure the uneven usage of synonymous codons into Cr. Let $FR'_{Cr}(c)$ be the relative probability in Cr of the codon c among all the synonymous codons to c, see (7). The probability $PR^i_{Cr}(C)$ to find C of frame i, into the Cr is computed as in (8). $LPR^i_{Cr}(C)$ is computed in a similar way of (4), and similar criteria are used to determine exons and introns.

$$FR'_{Cr}(c) = \frac{F'_{Cr}(c)}{FA'_{Cr}(c)} \tag{7}$$

$$PR^i_{Cr}(C) = \prod_{j=1}^{m} FR'_{Cr}(c^i_j); 1 \leq i \leq 3 \tag{8}$$

5 Design of a Methodology to Predict Exons and Introns

Our proposed methodology has the main objective to predict exons and introns in a DNA sequence by the introduced variants in Section 4. This methodology has the following steps:

1. Applying KM. Given two codon sequence C and Cr, such as $C \subset Cr$, apply KM to Cr to build a distribution of overlapped trigrams (see Table 3) of a DNA sequence Cr, to use it instead of the random distribution of the coding measure when the coding potential of the DNA sequence C be computed.

2. Determining coding potentials. Compute the coding potentials of the codon sequence C with the variants of coding measures proposed in Section 4.

In Fig. 4 the **Kasiski** procedure with the parameters Cr and N is shown. This procedure is the first step of KM and it is an exhaustive search of overlapped N-grams with a complexity of $O(n^2)$, in this case, $N = 3$ because we search for triplets in Cr. Where *length* is the number of nucleotides of Cr, in lines 3 and 8, $index_1$ and $index_2$ are used as indexes to compare the nucleotides along Cr. That means, for each $index_1$ from 1 to $length - N$ and for each $index_2$ from $index_1+1$ to $length-N$ are compared $Cr[index_1]$ with $Cr[index_2]$, $Cr[index_1 + 1]$ with $Cr[index_2 + 1]$ and $Cr[index_1 + 2]$ with $Cr[index_2 + 2]$, to know if $Cr[index_1 \dots index_1 + 2]$ and $Cr[index_2 \dots index_2 + 2]$ are the same triplet. There is a while loop in line 10, to find overlapped N-grams with the composed condition $Cr[index_1 + counter] = Cr[index_2 + counter]$ and $counter! = 0$, if it is satisfied to *true*, the *counter* is incremented, when $counter = N$ there are two N-grams starting in the positions $index_1$ and $index_2$. Then in line 13, the procedure **Search_and_Save_N-gram**$(index_1, index_2, Cr, N)$ is invoked to keep those N-grams and their positions $index_1$ and $index_2$. In lines 5-7, the procedure **Search_and_Save_N-gram**$(index_1, , Cr, N)$ without the second index is invoked to keep the last N-gram of Cr in the position $length - N$, in case of this N-gram has not been saved yet. Also, in lines 19-20, the same procedure without the second index is invoked to keep a N-gram that is not repeated along Cr because *unique* had never changed to false in line 14. It means that this N-gram is unique in Cr.

6 Experimental Results

The proposed methodology was applied to a set of 178 DNA sequences of homo sapiens [8], calculating their coding potentials in segments with size of 54, 108 and 162pb. Each sequence was preprocessed with the algorithm of Fig 4 which let calculate all the overlapped trigrams to create its non coding model of DNA for each variant. The sequences of [8] have known records of exons and introns. We carry out the proposed methodology with the new coding measures, if the coding potential of an evaluated segment is less than zero then we have an intron, otherwise is an exon. We set an error rate of 10% to the prediction of both exons and introns, this means that for prediction of exons in each sequence each we can fail 10% of the number of real exons, for introns we also set a 10% of error rate. The accuracy of the prediction of exons is the mean between the sensitivity (9) and specificity (10) of the prediction, similarly the accuracy of prediction of introns is computed.

Sensitivity of the exon prediction

$$S_n = \frac{NPEC}{NRE} \tag{9}$$

```
1 Kasiski(Cr, N)
2 begin
3 For index_1 = 1 to length − N
4     unique ← true
5     If index_1 = length − N
6         Call Search_and_Save_N-gram(index_1, ,Cr,N)
7     End If
8     For index_2 = index_1 + 1 to length − N
9         counter ← 0
10        While Cr[index_1 + counter] = Cr[index_2 + counter] and counter! = −1
11            Increment counter
12            If counter = N
13                Call Search_and_Save_N-gram(index_1,index_2,Cr,N)
14                unique ← false
15                counter ← −1
16            End If
17        End While
18    End For
19    If unique = true
20        Call Search_and_Save_N-gram(index_1, ,Cr,N)
21    End If
22 End For
23 end
```

Fig. 4. Pseudo-code of exhaustive search of Kasiski Method applied to Cr

Where $NPEC$=Number of Predicted Exons Correctly
and NRE=Number of Real Exons.

Specificity of the exon prediction

$$S_p = \frac{NPEC}{TNPE} \tag{10}$$

Where $NPEC$=Number of Predicted Exons Correctly
and $TNPE$=Total Number of Predicted Exons.

The prediction of exons becomes harder when the evaluated segment is smaller, in [5] the best measure was hexamer usage. The accuracy of the measure published in [5] and our results with the variants based on KM are shown in Table 4. The variant VPCKM improves the accuracy of the winner hexamer usage in the prediction of exons. The variants VCUKM is just under hexamer usage, but is over asymmetric position, hexamer-1,hexame-2 and the same CU. VAAUKM increase their accuracy respect to their originals but is under 5.4 percentiles points related to VPCKM. Also we report the first results of prediction of introns with an acceptable percentage in the mean accuracy for the three variants.

Table 4. Accuracy of prediction of coding measures published in [5] and variants based on KM. Mean between the specificity and sensitivity of the prediction.

	Prediction of Exons			Prediction of Introns
Coding Measure	Human 54	Human 108	Human 162	
VPCKM	**70.6**	**72.4**	**72.8**	**81.2**
Hexamer Usage	**70.5**	**73.1**	**74.2**	**X**
VCUKM	**70.4**	**70.7**	**71.2**	**82.7**
Asymmetric Position	70.2	76.6	80.6	**X**
Hexamer-1	69.9	72.6	73.8	**X**
Hexamer-2	69.9	72.6	73.8	**X**
UC	65.2	68	69.5	**X**
VAAUKM	**65.2**	**60.4**	**64.7**	**83.4**
UA	60.6	63.4	64.7	**X**
Prototype Codon	54.7	56.1	56.4	**X**

7 Conclusions

In this work, we propose a new methodology for the prediction of exons and introns based on a cryptanalysis method of Kasiski. We introduced three new coding measures based on codon biased coding measure and KM. The accuracy published by Fickett, with the winner hexamer usage for the prediction of exons, was improved with the variant VPCKM coding measure. Also the first results of the accuracy of prediction of introns were reported replacing in each selected measure the non coding model of DNA by a distribution which could be related with the tested sequence. The best coding measure was the VAAUKM variant with 83.4% of accuracy.

Like future work, we propose the adaptation of KM to the hexamer usage, with the goal to test its accuracy due that it was the winner of [5]. Also we propose these new measures to be incorporated to scientific software to improve the accuracy of the gene finding for example GENID, GENSCAN, and AUGUST.

References

[1] Griffiths, A., Gelbart, W., Lewontin, R., Miller, J.: Modern Genetic Analysis, 2nd edn (2002)
[2] Santos, A.: Criptoanalisis del Código Genético. Universidad de la Coruña. Tesis de Maestria (2000)
[3] Staden, R.: Codon preference and its use in identifying protein coding regions in long dna sequences. Nucleic Acids Research (1982)
[4] McCaldon, P., Argos, P.: Proteins: Structure, Function and Genetics 4, 99–122 (1988)
[5] Fickett, J.W., Tung, C.S.: Assessment of protein coding measures. Nucleic Acids Research 20, 6441–6450 (1992)

[6] Gribskov, M., Devereux, J., Burgess, R.B.: The codon preference plot: graphic analysis of protein coding sequences and prediction of gene expression. Nucleic Acids Research 12, 539–549 (1984)
[7] Gebbie, S.: A survey of the Mathematics of Cryptology. Masther's Thesis. University of the Witwatersrand, Johanesburg (2003)
[8] Software Predictor de Genes, http://augustus.gobics.de/

Chronic Hepatitis and Cirrhosis Classification Using SNP Data, Decision Tree and Decision Rule*

Dong-Hoi Kim[1], Saangyong Uhmn[1], Young-Woong Ko[1]
Sung Won Cho[2], Jae Youn Cheong[2], and Jin Kim[1]

[1] Department of Computer Engineering, Hallym University
Chuncheon, Kangwondo 200-702
Republic of Korea
{kdh,suhmn,yuko,jinkim}@hallym.ac.kr
[2] Genomic Research Center for Gastroenterology
School of Medicine, Ajou University
Suwon, Gyeonggi-do 443-749
Republic of Korea
{sungwoncho,jaeyoun620}@ajou.ac.kr

Abstract. A machine learning technique, decision tree, is used to predict the susceptibility to two liver diseases, chronic hepatitis and cirrhosis, from single nucleotide polymorphism(SNP) data. Also, it is used to identify a set of SNPs relevant to those diseases. The experimental results show that a decision tree is able to distinguish chronic hepatitis from normal with accuracy of 69.59% and cirrhosis from normal with accuracy of 76.72% and the C4.5 decision rule is with accuracy of 69.59% for chronic hepatitis and 79.31% for cirrhosis. The experimental results show that decision tree is a potential tool to predict the susceptibility to chronic hepatitis and cirrhosis from SNP data.

1 Introduction

A SNP is a variation in a single base pair, and is the most basic and common unit of genetic variation. It is distinguished from rare variations by a requirement for the least abundant allele to have a frequency of 1% or more. It is estimated that there are 3-10 million SNPs in the present human population, representing substantially less than 0.1% of the billions of base pairs that make up the human genome[1]. This tiny fraction of distinctive base pair changes accounts for virtually all difference between individuals, including medically significant differences like variations in disease susceptibility to or in response to medications. In 1992, The SNP Consortium(TSC)[2] was established to promote the identification of SNPs and to ensure that SNPs are kept in the public domain.

* This work was supported by a grant from the Korea Health 21 R&D Project, Ministry of Health and Welfare, Republic of Korea (A010383). Also, it was supported by Hallym University Research Fund, HRF-2004-40.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 585–596, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Chronic hepatitis is inflammation of the liver that lasts at least 6 months. Though much less common than acute hepatitis, chronic hepatitis can persist for years, even decades. In most people, it is quite mild and does not cause significant liver damage. However, In some people, continued inflammation slowly damages the liver, eventually producing cirrhosis (severe scarring of the liver), liver failure, and sometimes liver cancer. About one third of people with chronic hepatitis develop it after a bout of acute viral hepatitis. The two thirds of people develop the disease gradually without any obvious symptoms, although most cases of chronic hepatitis are still caused by one of the hepatitis viruses.

Cirrhosis is the destruction of normal liver tissue that leaves non-functioning scar tissue surrounding areas of functioning liver tissue. Most of the common causes of liver injury, when repeated or sustained over a long time, can result in cirrhosis. The scar tissue that forms impairs the liver's ability to function and obstructs the flow of blood through the portal vein (the vein that carries blood from the intestines to the liver). As a result of this obstruction, high blood pressure (portal hypertension occurs. In many parts of Asia and Africa, chronic hepatitis is a major cause of cirrhosis.

There is growing evidence that genetic variation plays an important role in the determination of individual susceptibility to complex disease traits. It has been thought that genetic associations may also provide a clue to the outcome to HBV infection. Several lines of evidence suggest that genetic polymorphisms have been reported to be involved in susceptibility to persistent HBV infection or development of HCC[3,4,5,6,7].

In this paper, we applied one of machine learning techniques, decision tree, to predict the susceptibility to chronic hepatitis and cirrhosis from SNPs of patients. A decision tree has been applied to many pattern recognition and data mining as well as biological data recently. With its inherent characteristic, decision tree performs feature selection while it builds a tree based on information gain or information gain ratio of each feature. We have applied four feature selection methods to change the order of features to be considered by the classifier and compared the results based on the accuracy. In addition, we have applied decision rule classifier to the same dataset to see the difference of those two classifying techniques, although decision rule classifier is built from decision tree classifier.

The remainder of the paper is organized as follows. In Section 2, we describe SNP, decision tree and decision rule as the background information. We state the problem and the proposed solutions in Section 3 and provide the experimental results in Section 4. we conclude in Section 5.

2 Backgrounds

2.1 SNP

SNPs are subtle variations, such as insertions, deletions and substitutions observed in the genomic DNA sequences of individuals of the same species[8]. In

diploid organisms, each DNA has two copies which are not completely identical. Each of two single copies is called a haplotype, while a description of data consisting of mixture of the two haplotypes is called a genotype. For complex diseases caused by more than a single gene, it is important to obtain haplotype data which identify a set of gene alleles inherited together. In haplotype description, only the position where the two copies are different is important, which is called a SNP. A SNP is a single nucleotide site where exactly two different nucleotides occur in a large percentage of the population. It is distinguished from rare variations by a requirement for the least abundant allele to have a frequency of 1% or more. There are several advantages of using SNP data over microarray data. First, SNPs are unlikely to change over time. That is, SNP pattern of a patient is likely to be the same as that collected at one's birth or early in life. Second, it can be collected from any tissue in the body so that it is relatively easy compared to the microarray which must be taken from the tumor tissue.

2.2 Decision Tree

In this paper, we propose to use decision tree for susceptibility prediction from SNP data. Because of huge amount of data, several machine learning techniques are used in the fields of human gene analysis. Among those techniques, decision trees are attractive for several reasons. First, due to the intuitive representation, the resulting model is easy to assimilate by humans[9,10]. Second, decision trees do not require any parameter setting from the user. And third, decision trees can be constructed relatively fast compared to other methods[11,12]. Last, the accuracy of decision trees is comparable to or superior to other methods[13,14]. Because of those characteristics, decision tree classifiers are used successfully in many diverse areas such as radar signal classification, character recognition, remote sensing, medical diagnosis, expert system, and speech recognition, to name only a few.

In addition, decision tree allows non-numerical values as well as numerical values for the feature values(or attributes), which is suitable for SNP analysis because each SNP is represented by a pair of nucleotides, A, C, T, and G.

The decision tree classifier is a set of hierarchical rules which are successively applied to the input data. Those rules are thresholds used to split the data into two or more groups. Each split is such that the descendant nodes contain more homogeneous data samples.

2.3 Decision Rule

In this paper, we use an implementation of decision tree classifier, C4.5 Release 8[15], for the experiment. We chose C4.5 because it is freely available and provides competitive results in several comparative studies[16,17]. C4.5 uses a greedy algorithm with one look-ahead step. In addition to the decision tree classifier, C4.5 generates a decision rule classifier from the tree it built. It computes the information contents of each condition attribute, and the results are used to

prune the condition attributes and create classification rules. A rule generated by C4.5 consists of simple predicates such as

$$If(a = 1)AND(b = 2)then(c = 4)[65.8\%] \quad (1)$$

A certainty value is assigned to each rule, which determines the confidence in that rule. In the previous example, the certainty value is 65.8%.

Even though the decision rules are generated from the decision tree, we obtain the difference experimental results compared to those by the decision tree classifier when we use the set of rules as the classifier.

3 Feature Selection and Methods

Because of the increased interest in SNPs in diagnosis and treatment of diseases, the number of collected SNPs are increasing so that the application of machine learning techniques to find the optimal set of SNPs is becoming impractical. Accordingly, it is required to apply feature selection techniques to reduce the number of SNPs to consider or the size of search space as with microarray expression data. One characteristic of decision tree is that it usually uses a subset of input features. However, it chooses the feature one by one based on the information gain or information gain ratio, which doesn't guarantee the optimal solution.

3.1 Problem Description

We have a set of n SNPs, $S = \{s_1, \ldots, s_n\}$, of m patients and their disease existence. We also have machine learning methods like decision tree which are able to predict disease existence for any given subset, $S' \subseteq S$. We define the prediction accuracy by a certain machine learning method for subset S' as $p(S', n)$. The number of possible subsets of S is $2^n - 1$. The goal is to find a subset S' which gives a maximal prediction accuracy, P_{max}. This problem is one of typical optimization problems.

$$\text{Select } S' \subseteq \{s_1, \ldots, s_n\} \text{ to maximize } p(S', n)$$

The simplest method to solve this problem is a brute force approach to evaluate all possible subsets and find the one giving the best prediction accuracy. It guarantees the subset with maximum accuracy. However, if the number of SNPs is large, this approach is intractable. This is one of the problems for which are believed that no efficient algorithm exists. This class of problems is called *NP-hard*[18]. In other words, the algorithms that are guaranteed to find an optimal solution for the problems with the size of n in reasonable time may not exist. To overcome this problem, we can use feature selection methods to select a small subset of SNPs for classification. Feature selection methods are not guaranteed to find the optimal solution, but that are very likely to find the near-optimal solution in reasonable time.

3.2 Feature Selection Algorithm

Incremental Information Gain(IIG). Information Gain[19] is a well known and empirically proven method for high-dimensional feature selection. It is the most well known criteria used by the decision tree. The information gain(IG) of a given attribute X with respect to the class attribute Y is the reduction in uncertainty about the value of Y when we know the value of X. We have used this value to construct n subsets of SNPs given to the classifier.

The following steps illustrate the algorithm. We assume S is a set of SNPs and an array P collects the performance measure(i.e. accuracy in this paper) of each set by the classifier.

INCREMENTAL-IG(S)

```
1  sort each SNP s_j ∈ S decreasing IG(s_j)
2  A = ∅
3  for i ← 1 to n
4        do A ← A ∪ {s_i} where s_i has the ith biggest IG
5           P[i] ← CLASSIFIER(A)
6  return P
```

Initially, all SNPs are sorted by IG and a SNP with the biggest IG is evaluated by the classifier. The search goes on in such a way that incorporating one SNP at a time to the SNP sets already evaluated. This algorithm generates only n sets of SNPs out of $2^n - 1$ possible subsets.

Forward Selection(FS). It starts with the empty set of SNPs. At each iteration, it selects one SNP from the set of unused SNPs whose inclusion gives the best prediction accuracy among the possible sets. The forward selection method proposes suboptimal but efficient way to incrementally select the best set of SNPs. The following illustrates this algorithm.

FORWARD_SELECTION(S)

```
1  A ← ∅
2  for i ← 1 to n
3        do  for j ← 1 to n − l + 1
4                  do  A' ← A ∪ {s_j}
5                      P[j] ← CLASSIFIER(A')
6            A ← A ∪ {s_k} where P[k] be the highest accuracy, for 1 ≤ k ≤ n−l+1
7            S ← S − {s_k}
8  return P, A
```

This algorithm evaluates $n + (n-1) + \cdots + 2 + 1 = \frac{n(n+1)}{2}$ sets out of $2^n - 1$ possible sets which is $\frac{n+1}{2}$ times as many sets as the incremental information gain.

Backward Elimination(BE). Contrary to forward selection, sequential backward elimination starts the search for the best set of SNPs with a solution with

all SNPs. At each iteration, one SNP is removed from the solution, until no further improvement in the quality of the solution can be attained.

```
BACKWARD_ELIMINATION(S)
1  A ← S
2  for i ← 1 to n
3        do  for j ← 1 to n − l + 1
4                 do  A′ ← A − {s_j}
5                     P[j] ← CLASSIFIER(A′)
6            A ← A − {s_k} where P[k] be the lowest accuracy, for 1 ≤ k ≤ n − l + 1
7            S ← S − {s_k}
8  return P, A
```

This algorithm evaluates $n + (n-1) + \cdots + 2 + 1 = \frac{n(n+1)}{2}$ sets out of $2^n - 1$ possible sets like forward selection. There are many variants of forward selection and backward elimination[20].

The Rule-based Feature Selection(RFS). In this paper, we use C4.5, an implementation of decision tree, for the experiment. Because of the characteristic of decision tree, it usually generates a tree with the subset of the input SNPs. From the analysis of the generated decision tree, decision rules and SNPs used, we found that in some cases a subset of the input SNPs were used by the generated rules even if a classifier was applied to the set of the previous execution. From this observation, we use it as a feature selection method. We repeat this procedure until there is no difference between the input SNPs to C4.5 and those used by the rules. It is illustrated as follows.

```
RULEBASEDFEATURESEL(S, D)
 1  Make decision rules R with S
 2  S′ ← SNPs used in R and
 3  a ← CLASSIFIER(S′,D)
 4  R′ ← R
 5  while (|S| > |S′|)
 6        do S ← S′
 7           Make decision rules R with S
 8           S′ ← SNPs used in R and
 9           a′ ← CLASSIFIER(S′,D)
10           if (a < a′)
11              then a ← a′
12                   R′ ← R
13  return S′, a
```

It starts with a set of SNPs S and at each iteration, at least one SNP is removed from the set so that thare are n sets at most out of $2^n - 1$.

4 Experimental Results

4.1 Data and Analysis Methods

Data. Table 1 shows the information on all SNPs and number of patients for the experiment. Our data set consists of 28 SNP data for 194 patients, and a class(healthy and chronic hepatitis) indicating the disease existence for chronic hepatitis. Another data set consists of 28 SNP data for 116 patients, and a class indicating the disease existence, (healthy and cirrhosis). The data is obtained by Ajou University Medical Center for Genomic Research Center for Gastroenterology Center[21]. The SNPs are selected from possible candidate genes which may occur chronic hepatitis and cirrhosis.

Table 1. Information of 28 SNPs and number of patients

Grp	No	SNPs	Seq.	Patients
1	1	CCR5(-2459)	G/A	
	2	RANTES(-403)	G/A	
	3	MCP1(-2518)	G/A	
	4	CCRR2-V64I	G/A	
	5	CXCR1-S276T	C/G	
	6	CXCR4-I138I	G/A	
2	7	IL1B-31	C/T	
	8	IL1B-511	C/T	
	9	ILIRN-S130S	C/T	
	10	IL1RN-3UTR	C/G	
	11	MBP-G54D	A/G	
3	12	IRF1(-410)	G/A	Hepatitis:control
	13	IFNGR2-Q64R	G/A	97:97
	14	IRF1(-388)	C/T	
	15	IL-10(-592)	A/C	Cirrhosis:control
	16	IL-10(-1082)	G/A	58:58
	17	IFNGR1(-56)	C/T	
	18	IFNGR1(+95)	C/T	
	19	IFNG(+874)	A/T	
	20	TNF-238	G/A	
	21	TNF-308	G/A	
4	22	IL18-S35S	C/A	
	23	MMP3-E45K	G/A	
	24	MMP3-D96D	C/T	
	25	MMP3-A362A	C/T	
	26	MMP9-R279Q	G/A	
	27	MMP9-Q688R	G/A	
	28	MMP9-G607G	C/A	

Classifier. We employed C4.5, an extension of the basic ID3 algorithm designed by Quinlan[15]. C4.5 generates a decision tree based on the training data and decision rules from the pruned tree. It also provides `c4.5rule` for the rule based classification. As stated above, we used these rules for feature selection.

Leave-one-out Cross Validation. Leave-one-out cross-validation(LOOCV) experiments are performed. We have 194 patients for chronic hepatitis(chronic hepatitis 97, healthy 97), and 116 patients for cirrhosis(cirrhosis 58, healthy 58). The C4.5 is trained using data from all but one of the patients[22]. The data not used in training is then assigned a class by the classifier. A single experiment consists of a series of leave-one-out experiments(194 times for chronic hepatitis and 116 for cirrhosis) because each SNP is held out and tested exactly once. The reason to use LOOCV is that it is very difficult to obtain clinical information of chronic hepatitis as well as cirrhosis patients.

Analysis Metrics. The experimental results are compared based on the following metrics. *A*ccuracy of a test can be expressed through sensitivity and specificity. *S*ensitivity refers to the ability of a certain diagnostic test to detect a chronic hepatitis(or cirrhosis). It is expressed as the probability of testing positive if the particular disease is truly present, i.e., the probability of having both positive test and a positive diagnosis. Therefore, a test with 90% sensitivity means that 90% of those with chronic hepatitis will test positive. *S*pecificity, on the other hand, refers to the probability of testing negative if chronic hepatitis is truly absent. Another parameter, accuracy is the ratio of the total of truly positive and truly negative over the total number of patients. Table 2 shows the calculation of sensitivity, specificity and accuracy.

Table 2. Calculation of accuracy, sensitivity and specificity

		Chronic hepatitis	
		+	−
test result	+	True Positive(TP)	False Positive(FP)
	-	False Negative(FN)	True Negative(TN)

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{2}$$

$$Sensitivity = \frac{TP}{(TP + FN)} \tag{3}$$

$$Specificity = \frac{TN}{(TN + FP)} \tag{4}$$

4.2 Accuracy of Chronic Hepatitis

We present the classification accuracy for chronic hepatitis using decision tree and rule with four feature selection algorithms in Figure 1. For comparison purpose, we have plotted that y-axis represents the accuracy, while the x-axis represents the number of SNPs in each set. As seen in Figure 1, the best accuracy is obtained using 4, 5 and 7 features, achieving 69.59% of correct classification(Leave-one-out cross-validation). Also, it shows that decision rule classifier and decision tree classifier provide the different experimental results, even though decision rules are generated from decision tree. In addition, the

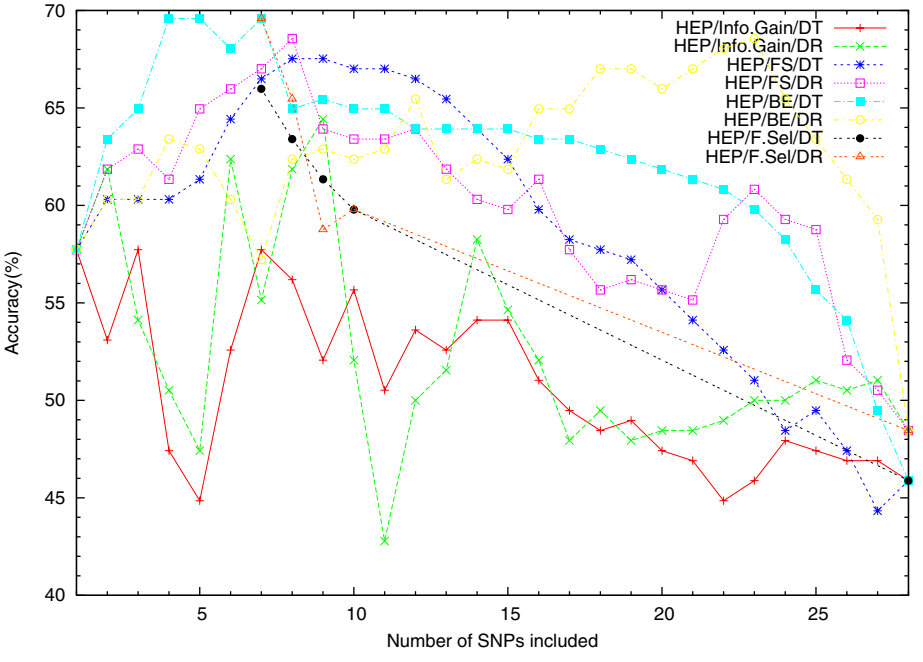


Fig. 1. Accuracy for hepatitis

decision rule classifier shows the better results for many subsets of SNPs than those by the decision tree classifier. However, the maximal accuracy is obtained by decision tree. In addition, the experiments with backward elimination and forward selection give the better results becuase these methods evaluate much more sets of SNPs so that the possiblility to find the optimal or near optimal solution is increased.

In Table 3, we present the accuracy, sensitivity and specificity for the each set with the highest accuracy obtained by four algorithms.

Table 3. HEP: Comparison of accuracy, sensitivity and specificity

Classifier	Algorithm	Accuracy	Sensitivity	Specificity
Decision Tree	IIG	57.76%	55.67%	59.79%
	FS	67.53%	72.16%	62.89%
	BE	69.59%	80.41%	58.76%
	RFS	69.59%	70.10%	69.07%
Decision Rule	IIG	64.43%	50.52%	78.35%
	FS	68.56%	68.04%	69.07%
	BE	68.56%	59.79%	77.32%
	RFS	69.59%	71.13%	68.04%

4.3 Accuracy of Cirrhosis

We present the classification accuracy for cirrhosis using decision tree and rule with four feature selection algorithms in Figure 2. It shows that the best accuracy is obtained using 6 features, achieving 79.31% of correct classification. Compared

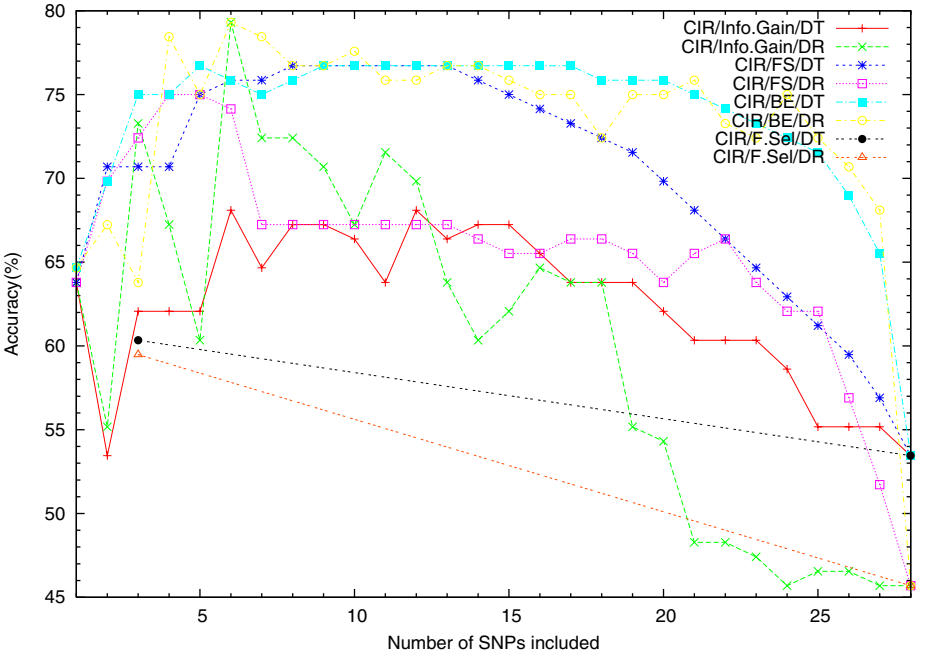


Fig. 2. Accuracy for cirrhosis

to the results in Figure **??**, it shows that the results by the tree are better than those by the rule classifier for many sets of SNPs and the maximal accuracy is obtained by the rule classifier.

Table 4. CIR: Comparison of accuracy, sensitivity and specificity

Classifier	Algorithm	Accuracy	Sensitivity	Specificity
Decision Tree	IIG	68.10%	68.97%	67.24%
	FS	76.72%	72.41%	81.03%
	BE	76.72%	74.14%	79.31%
	RFS	60.34%	63.79%	56.90%
Decision Rule	IIG	79.31%	75.86%	82.76%
	FS	75.00%	74.14%	75.86%
	BE	79.31%	86.21%	72.41%
	RFS	59.48%	55.17%	63.79%

In Table 4, we present the accuracy, sensitivity and specificity for the each set with the highest accuracy obtained by four algorithms.

5 Conclusion

We evaluate the possibility of machine learning techniques, decision tree and decision rule, as a tool to diagnose chronic hepatitis and cirrhosis. Also, we try to identify the related SNPs using four feature selection algorithms: incremental information gain, forward selection, backward elimination, and decision rule-based feature selection. The experimental results show that decision tree and rule can be potential tools for diagnosis of two liver diseases, chronic hepatitis and cirrhosis. However, due to the number of sets to evaluate, IIG and RFS may provide the better execution performance for larger datasets, whereas forward selection and backward elimination are applicable to the dataset with small number of SNPs.

However, the experimental results are not consistent for different dataset and with different feature selection algorithm. Although DT with BE and with RFS, and DR with RFS provide the highest accuracy for chronic hepatitis, DR with IIG and with RFS give the highest accuracy for cirrhosis. We think that it is because the characteristic of the dataset may affect the performance so that several different algorithms are needed to be applied to get the higher accuracy. We also think that those feature selection and elimination algorithms can not provide the optimal solution because of their greedy nature, which means that we have to use other algorithm to find the set of SNPs with the maximum accuracy.

References

1. Phillips, M., Boyce-Jacino, M.: A primer on SNPs - part 1. Innovations in Pharmaceutical Technology 1, 54–58 (2001)
2. TSC: The SNP consortium ltd. Website (1999), `http://snp.cshl.org`
3. Ahn, S.H., Han, K.H., Park, J.Y., Lee, C.K., Kang, S.W., Chon, C.Y., Kim, Y.S., Park, K., Kim, D.K., Moon, Y.M.: Association between hepatitis B virus infection and HLA-DR type in Korea. Hepatology 31, 1371–1373 (2000)
4. Ben-Ari, Z., Mor, E., Kfir, B., Sulkes, J., Tambur, A.R., Tur-Kaspa, R., Klein, T.: ytokine gene polymorphisms in patients infected with hepatitis B virus. American Journal of Gastroenterology 98, 144–150 (2003)
5. Höhler, T., Kruger, A., Gerken, G., Schneider, P.M., Meyer, K.H., Büschenfelde, Z., Rittner, C.: A tumour necrosis factor-alpha (TNF-α) promoter polymorphism is associated with chronic hepatitis B infection. Clinical and Experimental Immunology 111(3), 579–582 (1998)
6. Kim, Y., Lee, H., Yoon, J., Kim, C., Park, M., Kim, L., Park, B., Shin, H.: Association of TNF-α promoter polymorphisms with the clearance of hepatitis B virus infection. Human Molecular Genetics 12, 2541–2546 (2003)
7. Shin, H.D., Park, B.L., Kim, L.H., Jung, J.H., Kim, J.Y., Yoon, J.H., Kim, Y.J., Lee, H.S.: Interleukin 10 haplotype associated with increased risk of hepatocellular carcinoma. Human Molecular Genetics 12(8), 901–906 (2003)

8. Bell, J.I.: single nucleotide polymorphisms and disease gene mapping. Arthritis Research 4(3), S273–S278 (2002)
9. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA (1984)
10. Mehta, M., Agrawal, R., Rissanen, J.: SLIQ: A fast scalable classifier for data mining. In: 5th Intl. Conf. on Extending Database Technology (March 18–32, 1996)
11. Shafer, J.C., Agrawal, R., Mehta, M.: SPRINT: A scalable parallel classifier for data mining. In: Vijayaraman, T.M., Buchmann, A.P., Mohan, C., Sarda, N.L. (eds.) Proc. 22nd Int. Conf. Very Large Databases, pp. 544–555. Morgan Kaufmann, San Francisco (1996)
12. Gehrke, J., Ramakrishnan, R., Ganti, V.: Rainforest - a framework for fast decision tree construction of large datasets. In: Gupta, A., Shmueli, O., Widom, J. (eds.) Proc. 24th Intl. Conf. on Very Large Database, pp. 416–427 (1998)
13. Murthy, S.K.: On Growing Better Decision Trees from Data. PhD thesis. Johns Hopkins University, Baltimore, Maryland (1995)
14. Lim, T.S., Loh, W.Y., Shih, Y.S.: An empirical comparison of decision trees and other classification methods. Technical Report 979, Univ. of Wisconsin, Madison, WI (June 1997)
15. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993) `http://www.rulequest.com/Personal/`
16. Wang, Y.: Cancer classification using loss of heterozygosity data derived from single-nucleotide polymorphism genotyping arrays. In: Proceedings of the 28th IEEE International Conference of the Engineering in Medicine and Biology Society, New York CIty, New York, pp. 5864–5867 (2006)
17. Krishnan, V.G., Westhead, D.R.: A comparative study of machine-learning methods to predict the effects of single nucleotide polymorphisms on protein function. Bioinformatics 19(17), 2199–2209 (2003)
18. Papadimitriou, C.H.: Computational complexity. Addison Wesley, Reading (1993)
19. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
20. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: The Int'l Joint Conference on Artificial Intelligences, pp. 1137–1145 (1995)
21. AGCG: Ajou university medical center for genomic research center for gastroenterology. Website (2007), `http://www.agcg.re.kr/main.php`
22. Efron, B.: Bootstrap methods: Another look at the jackknife. The Annals of Statistics 7(1), 1–26 (1979)

Reconstruction of Suboptimal Paths in the Constrained Edit Distance Array with Application in Cryptanalysis*

Slobodan Petrović[1] and Amparo Fúster-Sabater[2]

[1] NISlab, Department of Computer Science and Media Technology,
Gjøvik University College, P.O. box 191, 2802 Gjøvik, Norway
slobodanp@hig.no
[2] Institute of Applied Physics (C.S.I.C.), Serrano 144, 28006 Madrid, Spain
amparo@iec.csic.es

Abstract. A new constrained edit distance computation algorithm is presented, as well as an algorithm for reconstruction of all the optimal and suboptimal paths in the array of partial constrained edit distances. The weight discrepancy of the suboptimal paths from the optimal ones does not overcome a threshold given in advance. The constraints include the maximum length of runs of deletions, the maximum length of runs of insertions and the total number of insertions. The edit operations are ordered in a sense that every substitution is preceded by at most one run of deletions followed by at most one run of insertions. This specific combination of constraints is determined by application of such a distance in cryptanalysis of pseudorandom sequence generators that employ decimation and interleaving of subgenerator sequences. Examples of generators from this family are the shrinking generator, the alternating step generator, the step 1/step 2 generator and so on. The search for the clock control sequence in such a generator is directed by relating the weight-difference threshold with the level of noise in the statistical model of the generator. In such a way, the efficiency of clock control sequence reconstruction is significantly improved. We show experimentally, on a problem of realistic size, that reconstruction of the clock control sequence in a generator from this family is feasible up to a relatively high level of noise in its statistical model.

Keywords: String Editing, Constrained Edit distance, Cryptanalysis, Dynamic Programming, Directed Search.

1 Introduction

Edit distance is a widely used measure in coding, computational biology, speech recognition and many other fields. It is defined as the minimum sum of elementary edit distances associated with substitutions, deletions and insertions needed to transform one string into another. Due to application particularities, certain constraints are often included in the general definition, such as the total number of deletions and/or insertions, the maximum lengths of runs of deletions and insertions , the total number of runs of deletions and/or insertions and so on [13].

* Work partially supported by Ministerio de Ciencia y Tecnología (Spain) under grants SEG 2004-02418 and SEG 2004-04352-C04.03.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 597–610, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Edit distance with or without constraints can be computed in an iterative manner, by filling the edit distance array. Then it is possible to reconstruct an optimal sequence of elementary edit operations (so called edit sequence) corresponding to every element of this array. There are two basic problems related with this reconstruction:

1. In general, the optimal edit sequence corresponding to an element of the edit distance array is not unique. Sometimes, we are interested in enumerating all the possible optimal edit sequences corresponding to an element of the edit distance array.
2. There is often a need to reconstruct not only the optimal edit sequences but also the suboptimal ones, where the maximum discrepancy from the optimum is given in advance. For example, in molecular biology the optimal edit sequences that transform a part of a DNA sequence to a part of another DNA sequence are often unrealistic, i.e. they do not correspond to the underlying transformation that really occurs in nature.

In this paper, we define an algorithm that reconstructs all the optimal and suboptimal edit sequences in the constrained edit distance array. The weight-difference of a suboptimal edit sequence from an optimal one does not overcome a threshold given in advance. The constraints deal with the permitted total number of insertions, the maximum lengths of runs of deletions and insertions and the ordering of elementary edit operations.

The constrained edit distance computation has not been treated extensively in the literature. The often used general dynamic programming approach can be adjusted to take into account the constraints related with the maximum lengths of runs of deletions and insertions [13]. However, the time and memory complexity of the computation of the constrained edit distance can be reduced compared to that achieved in [13]. For example, in [12], a different more efficient approach to computing constrained edit distance is used, but only the constraints on the total number of insertions are included, not the constraints on the run lengths of deletions/insertions.

Our constrained edit distance computation algorithm uses the approach from [12] as the basis, but with deletion/insertion run length constraints included, as well as ordering of edit operations. We also define an algorithm that searches for all the optimal and suboptimal paths in the constrained edit distance array. This search can be directed by gradually increasing the weight-difference threshold.

The specific combination of constraints used here is motivated by the application of such an edit distance measure in cryptanalysis of pseudorandom sequence generators with decimation and interleaving of subgenerator sequences. The examples of such generators are the shrinking generator [4], the alternating step generator [8], the step 1/step 2 generator [3], and so on. We apply the defined optimal and suboptimal path reconstruction procedure in the clock control sequence reconstruction in such generators.

In addition to directing of the search by gradually increasing the weight-difference threshold, we additionally reduce the time and space complexities of the suboptimal path enumeration, compared to the complexities of general path enumeration procedures (the K-best path algorithm [5] or computational biology specific algorithms [11]), by taking into account the very definition of the constrained edit distance. Namely, the number of branchings from a cell in the constrained edit distance array is limited, and this limit depends on the constraints included in the edit distance definition.

We experimentally show that a cryptanalytic algorithm in which such a path enumeration procedure is applied is feasible up to a relatively high noise level in the statistical model of the analyzed generator. The example used in the experiments was of realistic size - the lengths of linear feedback shift registers (LFSRs) employed were up to 30, which can be met in, for example, cryptographic protection algorithms from the A5 family used in the GSM mobile telephony [1].

2 Constrained Edit Distance Algorithm

Given arbitrary strings X and Y of finite lengths N and M, respectively, over a finite alphabet $\mathcal{A}$, we deal with the problem of transforming the string X to Y by means of the elementary edit operations of deletion, insertion, and substitution, under the following constraints:

C1 The number of insertions belongs to a predefined set T.
C2 The maximum lengths of runs of deletions and insertions are F and G, respectively.
C3 The edit sequence is ordered in a sense that every substitution is preceded by at most one run of deletions followed by at most one run of insertions.

The constrained edit distance $D(X, Y)$ is then defined as the minimum sum of elementary edit distances associated with the edit operations of deletion, insertion, and substitution needed to transform X to Y, subject to the assumed constraints.

We first define elementary distances associated with the edit operations of deletion, insertion, and substitution of symbols. Along with the alphabet $\mathcal{A}$, we introduce the "empty" symbol ϕ reserved for the presentation of operations of deletion and insertion. Let $\mathcal{A}^* = \mathcal{A} \cup \{\phi\}$. Nonnegative real-valued elementary edit distances are defined in the following way:

1. $d(a, \phi)$ is the elementary distance associated with the deletion of a symbol $a \in \mathcal{A}$;
2. $d(\phi, b)$ is the elementary distance associated with the insertion of a symbol $b \in \mathcal{A}$;
3. $d(a, b)$ is the elementary distance associated with the substitution of the symbol a with the symbol b, where $a, b \in \mathcal{A}$.

We also define the compression operator C in the following way: for an arbitrary finite length string Z over $\mathcal{A}^*$, $C(Z)$ stands for the string over $\mathcal{A}$ that is obtained from Z by removing all the empty symbols from it (see [12], for example).

Every editing transformation as a sequence of elementary edit operations can be equivalently represented in an ordered way according to the constraint C3 defined above. We call such a representation an *edit sequence* and denote it by $S = (X', Y')$, where $(X', Y') \in \mathcal{A}^* \times \mathcal{A}^*$, and for every (X', Y') the following rules hold:

1. $C(X') = X, C(Y') = Y$;
2. $\mid X' \mid = \mid Y' \mid = L$, where $\max\{N, M\} \leq L \leq N + M$;
3. for all i, $1 \leq i \leq \mid X' \mid$, the variables x'_i and y'_i cannot simultaneously take the value ϕ;
4. (X', Y') satisfies the constraints C2 and C3 introduced above.

By $\mathcal{G}(X,Y)$ we denote the set of all the permitted edit sequences S that transform the string X to the string Y, i.e. $\mathcal{G}(X,Y) = \{(X',Y') \mid (X',Y')$ satisfies the constraints C2 and C3 $\}$. Also, let $\mathcal{G}_T(X,Y)$ denote the set of all the edit sequences from $\mathcal{G}(X,Y)$ that satisfy the constraint C1 as well. There is an one-to-one correspondence between the set $\mathcal{G}(X,Y)$ of all the permitted edit sequences (X',Y') and the set of all the permitted editing transformations of X to Y. Because of that, for the constrained edit distance we have the following expression:

$$D(X,Y) = \min\left\{\sum_{i=1}^{L} d(x'(i),y'(i)) \mid (X',Y') \in \mathcal{G}_T(X,Y)\right\}, \tag{1}$$

where $L = N + M - s$ and $0 \le s \le \min\{N,M\}$.

In order to compute the constrained edit distance efficiently, following the concepts from [12], we introduce a new quantity, the *partial constrained edit distance* $W(i,e,s)$ as a constrained edit distance between the prefix X_{e+s} of the sequence X and the prefix Y_{i+s} of the sequence Y. More precisely, let $\mathcal{G}_{i,e,s}$ denote the set of all the edit sequences from $\mathcal{G}(X_{e+s},Y_{i+s})$ that transform X_{e+s} to Y_{i+s} using exactly i insertions, e deletions, and s substitutions. A triple (i,e,s) is feasible if $\mathcal{G}_{i,e,s}$ is not empty. Then

$$W(i,e,s) = \min\left\{\sum_{j=1}^{i+e+s} d(x'(j),y'(j)) \mid (X'_{e+s},Y'_{i+s}) \in \mathcal{G}_{i,e,s}\right\}. \tag{2}$$

It is easy to verify that

$$D(X,Y) = \min_{i\in T} W(i,N-M+i,M-i). \tag{3}$$

where each i from T automatically satisfies the inequality $\max\{0,M-N\} \le i \le M$.

Note that the set of feasible values of (i,e,s) is, due to the assumed constraints, given by

$$0 \le s \le \min\{N,M\},\ \ 0 \le e \le \min\{N-s,sF\},\ \ 0 \le i \le \min\{M-s,sG\}. \tag{4}$$

Namely, a triple (i,e,s) is feasible if and only if it satisfies (4).

The following theorem enables efficient computation of the constrained edit distance $D(X,Y)$.

Theorem 1
Under the combination of the constraints C1,C2, and C3, the partial constrained edit distance $W(i,e,s)$ satisfies the recursion:

$$W(i,e,s) = \min_{(i_1,e_1)\in Q}\left\{W(i-i_1,e-e_1,s-1) + \sum_{j=1}^{e_1} d(x(e+s-e_1+j),\phi) + \right.$$
$$\left. + \sum_{j=1}^{i_1} d(\phi,y(i+s-i_1+j)) + d(x(e+s),y(i+s))\right\}, \tag{5}$$

$1 \le s \le \min\{N, M\}, 1 \le e \le \min\{N - s, sF\}, 1 \le i \le \min\{M - s, sG\}$, where Q is the set of all the pairs (i_1, e_1) such that

$$\max\{0, i - \min\{M - s, (s-1)G\}\} \le i_1 \le \min\{G, i\} \tag{6}$$

$$\max\{0, e - \min\{N - s, (s-1)F\}\} \le e_1 \le \min\{F, e\}. \tag{7}$$

For $s = 0$, $0 \le e \le \min\{N, F\}$, and $0 \le i \le \min\{M, G\}$,

$$W(i, e, 0) = \sum_{j=1}^{e} d(x(j), \phi) + \sum_{j=1}^{i} d(\phi, y(j)). \tag{8}$$

Proof

For $s = 0$ the proof is trivial. Assume that $s \ge 1$. We start from the basic expression (2) for $W(i, e, s)$. We partition the set $\mathcal{G}_{i,e,s}$ according to the sequences of deletions and insertions before the last substitution; namely, let $\mathcal{G}^q_{i,e,s}$, $q = (i_1, e_1)$ denote the set of all the sequences in $\mathcal{G}_{i,e,s}$ that end with e_1 deletions and i_1 insertions, before the last substitution, for $0 \le e_1 \le \min\{F, e\}$, $0 \le i_1 \le \min\{G, i\}$. $\mathcal{G}^q_{i,e,s}$ is empty iff $(i - i_1, e - e_1, s - 1)$ is not feasible. Therefore, in view of the equation (4), we obtain that $\mathcal{G}^q_{i,e,s}$ is not empty iff equations (6) and (7) are true. Accordingly, we partition the set $\mathcal{G}_{i,e,s}$ into the subsets $\mathcal{G}^q_{i,e,s}$, $q \in Q$. Rewriting Equation (2) and taking out of the inner minimization term the terms that are constant when the set $\mathcal{G}^q_{i,e,s}$ is given, we get

$$W(i, e, s) = \min_{q \in Q} \left\{ \sum_{j=1}^{e_1} d(x(e + s - e_1 + j), \phi) + \sum_{j=1}^{i_1} d(\phi, y(i + s - i_1 + j)) + \right.$$

$$+ d(x(e + s), y(i + s)) +$$

$$\left. + \min \left\{ \sum_{j=1}^{i+e+s-i_1-e_1-1} d(x'(j), y'(j)) \mid (X'_{e+s}, Y'_{i+s}) \in \mathcal{G}^q_{i,e,s} \right\} \right\}. \tag{9}$$

Since every edit sequence in $\mathcal{G}^q_{i,e,s}$ ends in one substitution preceded by exactly e_1 deletions and i_1 insertions, and the constraints are on the numbers of consecutive deletions and insertions, it follows that an edit sequence (X'_{e+s}, Y'_{i+s}) belongs to $\mathcal{G}^q_{i,e,s}$ iff its prefix $(X'_{e-e_1+s-1}, Y'_{i-i_1+s-1})$ belongs to $\mathcal{G}_{i-i_1,e-e_1,s-1}$, for any $q \in Q$. Therefore, considering Equation (2), we have

$$\min \left\{ \sum_{j=1}^{i+e+s-i_1-e_1-1} d(x'(j), y'(j)) \mid (X'_{e+s}, Y'_{i+s}) \in \mathcal{G}^q_{i,e,s} \right\} =$$

$$= \min \left\{ \sum_{j=1}^{i+e+s-i_1-e_1-1} d(x'(j), y'(j)) \mid (X'_{e-e_1+s-1}, Y'_{i-i_1+s-1}) \in \mathcal{G}_{i-i_1,e-e_1,s-1} \right\} =$$

$$= W(i - i_1, e - e_1, s - 1). \tag{10}$$

Consequently, (9) is reduced to (5).

□

3 Reconstruction of Optimal and Suboptimal Paths

We call *optimal paths* the paths through the edit distance array that begin at $W[i, N-M+i, M-i]$, $i \in T$, for which the constrained edit distance corresponding to $W[i, N-M+i, M-i]$ gets the minimum value. Let $pl \leq \min\{N, M\}$ be the length of the prefix of the optimal/suboptimal path to be reconstructed. Because of the constraint C3 defined above regarding ordering of edit operations, all the optimal paths pass through the cells $W[i_{p_1}, e_{p_1}, pl], \ldots, W[i_{p_n}, e_{p_n}, pl]$ in the layer $s = pl$ of the array W, where n depends on the particular sequences.

To determine the points in the layer $s = pl$ through which the optimal paths pass we assign to every cell $W[i, e, s]$, besides the value $W[i, e, s].c$ of the corresponding constrained edit distance, four associated vectors:

1. The vector of 'primary' pointers vp to the cells $W[vp[1], s-1], \ldots, W[vp[k], s-1]$ from which it is possible to arrive to the cell $W[i, e, s]$ with the minimum weight increment, where $vp[j] = (i[j], e[j])$, $j = 1, \ldots k$.
2. The vector of 'updated' pointers vu to the cells $W[vu[1], pl], \ldots, W[vu[l], pl]$, through which it is possible to arrive to the cell $W[i, e, s]$ with the minimum weight increment (see for example [9]), where $vu[j] = (i[j], e[j])$, $j = 1, \ldots l$.
3. The vector of pointers ve to the cells $W[ve[1], s-1], \ldots, W[ve[m], s-1]$ from which it is possible to arrive to the cell $W[i, e, s]$ regardless of the weight increment, where $ve[j] = (i[j], e[j])$, $j = 1, \ldots m$.
4. The vector of values vj of the edit distances corresponding to the elements of the vector ve. The cardinality of this vector is also m.

The actual values of k, l, and m depend on the particular sequences. The array W is filled by means of the Algorithm 1, which implements the equation (5) together with the updating of the four vectors mentioned above. The Algorithm 1 is a dynamic programming algorithm - the contents of any cell of the constrained edit distance array is obtained by means of previously computed contents of some other cells of the same array.

Algorithm 1
Input:

- The sequences X and Y of lengths N and M, respectively.
- The necessary length pl of the optimal/suboptimal path to be reconstructed.
- The maximum lengths F and G of runs of deletions and insertions, respectively.
- The elementary distances d_e and d_i associated with the deletion and the insertion of a symbol, respectively.
- The elementary edit distance $d[x, y]$ associated with the substitution of the symbol x with the symbol y, $\forall(x, y)$.

Output:

- The array W of edit distances with the vectors vp, vu, vj, and ve associated with every cell.

comment Initialization

$s_{\max} \longleftarrow \min\{N, M\}$; $e_{\max} \longleftarrow \max\{\min\{N - s, s * F\}\}$, $s = 1, \ldots, s_{\max}$;

$i_{\max} \longleftarrow \max\{\min\{M - s, s * G\}\}$, $s = 1, \ldots, s_{\max}$;

$W[i, e, s].c \longleftarrow \infty$, $i = 0, \ldots, i_{\max}$, $e = 0, \ldots, e_{\max}$, $s = 0, \ldots, s_{\max}$; the vectors vp, vu, vj, and ve associated with every cell $W[i, e, s]$ are empty.
$W[0, 0, 0].c \longleftarrow 0$;

comment Main loop

for $s \longleftarrow 1$ **until** $s_{\max}$ **do**
begin
for $e \longleftarrow 1$ **until** $\min\{N - s, s * F\}$, $i \longleftarrow 1$ **until** $\min\{M - s, s * G\}$ **do**
begin

Let q be the minimum value of the expression
$W[i - i_1, e - e_1, s - 1].c + e_1 * d_e + i_1 * d_i + d[X[e + s], Y[i + s]]$,
$e_1 = \max\{0, e - \min\{N - s, (s - 1) * F\}\}, \ldots, \min\{e, F\}$,
$i_1 = \max\{0, i - \min\{M - s, (s - 1) * G\}\}, \ldots, \min\{i, G\}$. (11)

Let n_q be the number of values of pairs (i_1, e_1) for which the expression (11) takes the value q. Then

$W[i, e, s].c \longleftarrow q$; $W[i, e, s].k \longleftarrow n_q$;

$W[i, e, s].vp$ is filled with n_q values of the expression $(i - i_1, e - e_1)$ corresponding to the values (i_1, e_1) for which the expression (11) takes the value q.
$W[i, e, s].vj$ is filled with all the values of the expression (11).
$W[i, e, s].ve$ is filled with the values $(i - i_1, e - e_1)$ corresponding to the values of $W[i, e, s].vj$.

end ;

comment Determining updated pointers vu.

if $s = pl + 1$ **then**
$W[i, e, s].vu \longleftarrow W[i, e, s].vp$,
$e = 0, \ldots, \min\{N - s, s * F\}$, $i = 0, \ldots, \min\{M - s, s * G\}$;
else if $s > pl + 1$ **then**

For every element of $W[i, e, s].vp$, $e = 0, \ldots, \min\{N - s, s * F\}$, $i = 0, \ldots, \min\{M - s, s * G\}$, the elements of $W[W[i, e, s].vp[i], s - 1].vu$, $i = 1, \ldots, W[i, e, s].k$ are placed into $W[i, e, s].vu$, deleting the repeated ones. Note that $W[i, e, s].vp[i]$ is two-dimensional and thus covers both the i and the e dimension in $W[W[i, e, s].vp[i], s - 1].vu$.

end.

Remark: The $[i, 0, s]$ layer, as well as the $[0, e, s]$ layer of the array W are initialized by executing the main loop procedure without unnecessary elements. □

Having constructed the edit distance array, the next step is reconstructing the optimal/suboptimal paths in it. From now on, by *paths* we mean fragments of paths that start in the layer $s = pl$ of the array W. There are three sets of paths to be reconstructed. The first one consists of optimal paths that start at the points $(i_{p_j}, e_{p_j}) = W[i, N - M + i, M - i].vu[j], j = 1, \ldots, W[i, N - M + i, M - i].l$, for those i for which $W[i, N - M + i, M - i].c$, $i \in T$ reaches its minimum value. The second one consists of suboptimal paths, whose weight-difference from the optimal ones is $\leq \mathcal{D}$, that start at the same points as the paths from the first set. The third set consists of suboptimal paths, whose weight-difference from the optimal ones is $\leq \mathcal{D}$, that start at other points in the layer $s = pl$.

The elements of the vector $W[i, N-M+i, M-i].vu, i \in T, W[i, N-M+i, M-i].c$ is minimum, at the end of the execution of the Algorithm 1 represent the initial points of the search for the elements of the first and second set mentioned above. As for the third set, if $\mid W[i_{p_j}, e_{p_j}, pl].c - W[i, e, pl].c \mid \leq \mathcal{D}$, $e = 0, \ldots, \min\{N - s, s * F\}$, $i = 0, \ldots, \min\{M - s, s * G\}$, $i \neq i_{p_j}$ $e \neq e_{p_j}$, for at least one j, then the point $W[i, e, pl]$ is an initial point of the depth first search for the paths of the third set.

The following algorithm is used to determine the optimal and suboptimal paths that start at every initial point $\mathcal{E}$ of any of the three sets:

Algorithm 2
Input:

- The array W of edit distances, obtained by means of the Algorithm 1.
- The values of pl, $\mathcal{E}$ and $\mathcal{D}$. Note that $\mathcal{E}$ is two-dimensional. It covers the i and e dimension of the array W.

Output:

- All the paths that start at the point $W[\mathcal{E}, pl]$ that belong to the corresponding set(s) (see text).

comment The $stack$ consists of the elements of the array W together with their coordinates; the cardinality of the $stack$ is nsp; the current length of the reconstructed path of the edit sequence is t.

comment Initialization

$t \longleftarrow 0$; $(i, e) \longleftarrow \mathcal{E}$; $s \longleftarrow pl$; $nsp \longleftarrow 0$;

comment Main loop

repeat
  $badpath \longleftarrow$ **false** ; **comment** This is the path overweight indicator
  while $((i > 0)$ **or** $(e > 0)$ **or** $(s > 0))$ **and not** $badpath$ **do**
    begin
      if $W[i, e, s]$ is a branching point **then**

        comment Put i, e, s and $W[i, e, s]$ on the $stack$

          $nsp \longleftarrow nsp + 1$; $stack[nsp] \longleftarrow (W[i, e, s], i, e, s)$;

comment Process a branching point

$badpath \longleftarrow$ **false** ;

repeat
Consider the possibility of branching from the current branching point to one of the possible successors, i.e. the point j. If this possibility is chosen, and after that only the branchings to the points that lead to the optimal subpaths are followed, then the total weight of the chosen subpath is
$aw \longleftarrow stack[nsp].W.vj[stack[nsp].W.j]$;
and the total weight of the corresponding path is
$tw \longleftarrow weight(\alpha, \beta, t) + aw$;
where $weight$ is the function that returns the weight of the path before the branching and (α, β) is the prefix of the edit sequence of length t.
comment $(iprev, eprev)$ is the value of (i, e) that corresponds to the previous path element
if $tw <= W[\mathcal{E}, pl].c + \mathcal{D}$ **then**
$(iprev, eprev) \longleftarrow stack[nsp].W.ve[stack[nsp].W.j]$;
$stack[nsp].W.j \longleftarrow stack[nsp].W.j - 1$;
if $stack[nsp].W.j = 0$ **then** $nsp \longleftarrow nsp - 1$;
until ($(iprev, eprev)$ was initialized from the $stack$) **or** (all the successors were examined) ;

if $(iprev, eprev)$ was not initialized from the $stack$ **then** $badpath \longleftarrow$ **true** ;

comment Process a non-branching point

if ($(iprev, eprev)$ was not initialized from the $stack$) **and** (**not** badpath) **then**
begin
$aw \longleftarrow W[i, e, s].vj[W[i, e, s].j]$; $tw \longleftarrow weight(\alpha, \beta, t) + aw$;
if $tw <= W[\mathcal{E}, pl].c + \mathcal{D}$ **then**
$(iprev, eprev) \longleftarrow W[i, e, s].ve[W[i, e, s].j]$;
else
$badpath \longleftarrow$ **true** ;
end ;

comment Reconstruct the current path element

if not $badpath$ **then**
begin
comment Substitution
$t \longleftarrow t + 1$; $\alpha[t] \longleftarrow X[e + s]$; $\beta[t] \longleftarrow Y[i + s]$;
for $ii \longleftarrow 1$ **until** $(e - eprev)$ + $(i - iprev)$ **do**
begin
$t \longleftarrow t + 1$;
if $ii <= i - iprev$ **then**
begin comment The run of insertions
$\alpha[t] \longleftarrow \varnothing$; $\beta[t] \longleftarrow Y[i + e + s - ii]$;
end

```
                else
                  begin comment The run of deletions
                    α[t] ⟵ X[i + e + s − ii] ; β[t] ⟵ ∅ ;
                  end
              end ;
            i ⟵ iprev ; e ⟵ eprev ; s ⟵ s − 1 ;
          end ;
      end ;
    if not badpath then
      Store the obtained edit sequence ;

  comment Back to the current branching point

    if nsp > 0 then
      begin
        t ⟵ t − stack[nsp].i − stack[nsp].e − stack[nsp].s ;
        (i, e, s) ⟵ stack[nsp].(i, e, s) ;
      end
until nsp = 0 .
```

□

4 Application in Cryptanalysis

We apply the algorithms 1 and 2, presented in the previous section, in the cryptanalysis of pseudorandom sequence generators with decimation and interleaving of subgenerator sequences. An example of such a generator is a generator with so called irregular clocking (the shrinking generator [4], the alternating step generator [8], the step 1/step 2 generator [3], etc.) Consider a pseudorandom sequence generator that contains an LFSR whose clock control sequence is produced by a general type subgenerator. If a sufficiently long prefix of the output sequence of such a generator is known, it is possible to determine a set of candidate initial states of the LFSR, which could generate the intercepted output sequence [7]. The statistical model used in [7] (Fig. 1) employs the edit distance with two elementary edit operations (deletions and substitutions) and with the constraint on the maximum length of runs of deletions.

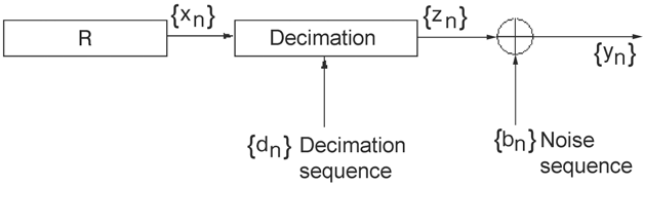


Fig. 1. The statistical model of the analyzed generator

After the attack [7] is performed as the first phase of the overall attack, a new phase of the attack should reconstruct the clock control sequence that, together with one of the candidate initial states of the LFSR obtained in the first phase, could generate the intercepted sequence. From the reconstructed clock control sequence, it is possible to reconstruct the initial state of the clocking subgenerator. We propose the use of algorithms 1 and 2 in this second phase of the attack.

The cryptanalyst possesses M consecutive bits of the modulo 2 sum $\{y_n\}$ of the decimated sequence $\{z_n\}$ and the noise sequence $\{b_n\}$ (Fig. 1). His/her task is to determine the initial state of the generator that produced the M intercepted bits of the sequence $\{y_n\}$.

We consider the ciphertext only attack on such a generator, which is a more realistic attack scenario than the known plaintext attack scenario used in the majority of previously defined reconstruction procedures [2,6,10], since it takes into account the influence of noise. We incorporate the algorithms 1 and 2 into a deterministic method of reconstruction of clock control sequences, in which the noise level is related with the permitted weight deviation from the optimum path weight used in the search process. The paths in the two dimensional edit distance array that correspond to candidate clock control sequences are reconstructed. By starting with the reconstruction of paths whose weight deviation from the optimum is 0 (the optimal paths - without noise) and then by increasing this weight deviation according to the noise level (the suboptimal paths), we make our search a directed one.

Note that the edit distance array used in the algorithms 1 and 2 is two dimensional (the i dimension is not needed, since there are no insertions in the statistical model of the generator), which significantly reduces computational complexity of the attack.

5 Experimental Results

From the cryptanalytic point of view, the number of paths necessary to find the clock control sequence should be as small as possible. This number depends on $\mathcal{D}$. Given a certain level of noise in the statistical model, the behaviour of the maximum value of $\mathcal{D}$, denoted by $\mathcal{D}_{max}$, was analyzed experimentally.

The experiment was carried out in the following way: 1000 initial states of the Step 1 / step 2 generator (a structure with two LFSRs, see Fig. 2) were chosen at random. In this structure, one LFSR, R_1, generates the binary (hence step 1 / step 2) clock control sequence for the other, R_2.

For each of the 1000 initial states, the output sequence corrupted by the noise sequence generated at random was produced. The noise level p was the control variable of the experiment. The set of candidates for the initial state of R_2 was determined. Once

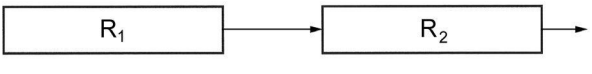


Fig. 2. The Step 1 / step 2 generator

the candidates were obtained, for a fixed value of $\mathcal{D}$, the optimal and suboptimal paths were determined from the edit distance matrix corresponding to each of them. This process was repeated starting from $\mathcal{D} = 0$ and incrementing the value of $\mathcal{D}$ until the clock control sequence generated by R_1 was found. The maximum value $\mathcal{D}_{max}$ obtained in this process was stored. At the end of the experiment, the mean value $\overline{\mathcal{D}_{max}}$ over the values of $\mathcal{D}_{max}$ obtained in every case was computed. The dependence of $\overline{\mathcal{D}_{max}}$ on p for different values of pl is depicted in the Fig. 3.

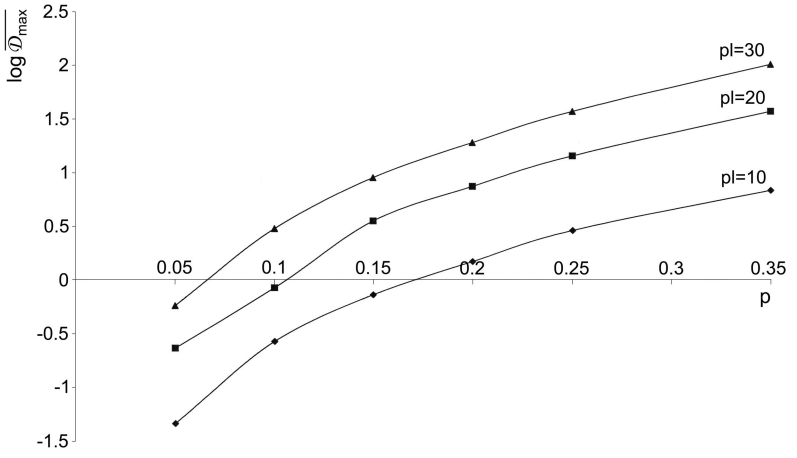


Fig. 3. Dependence of $\overline{\mathcal{D}_{max}}$ on p

From the Fig. 3, it can be observed that:

1. For $p = 0$, only the optimal paths that start in the column pl of the edit distance matrix need to be reconstructed. For low levels of noise, the value of $\overline{\mathcal{D}_{max}}$ is small, which means that the number of reconstructed paths in the edit distance matrix is small.
2. $\overline{\mathcal{D}_{max}}$ depends approximately linearly on pl. This makes our attack possible to use against the schemes that employ relatively long LFSRs.
3. The dependence of $\overline{\mathcal{D}_{max}}$ on p is also approximately linear, which makes our attack feasible even for relatively high levels of noise. This noise may be a consequence of using other devices whose output sequence is added modulo 2 to the output of the analyzed structure.

The experiment was performed on an ordinary office computer and for LFSR lengths up to 30 and the noise level up to 0.35 it was practically feasible on such a machine. The examined LFSR lengths were of practical importance. For example, such LFSRs have been widely deployed in the GSM mobile telephony cryptographic protection algorithm family A5 (A5/1 and A5/2) [1].

6 Conclusion

In this paper, a new algorithm for constrained edit distance computation, as well as an algorithm for suboptimal paths reconstruction in the constrained edit distance array are presented. The weight discrepancy of the suboptimal paths from the optimal ones is given in advance. The elementary edit operations of deletions, insertions, and substitutions are ordered in a sense that every substitution is preceded by at most one run of deletions, followed by at most one run of insertions. The maximum lengths of runs of deletions and insertions represent additional constraints. The procedure of suboptimal paths reconstruction is applied in the cryptanalysis of a family of pseudorandom generator schemes containing irregularly clocked LFSRs. The influence of noise is taken into account by relating the noise level with the permitted deviation from the noiseless-case path weight in the search process. The statistical model of the basic structure of this family of generators that employs the constrained edit distance is used. The search in the edit distance array is directed because it starts with the reconstruction of the paths with zero deviation from the noiseless case path weight and then iteratively increments this deviation. The maximum value of weight deviation necessary for the reconstruction of the actual clock control sequence depends on the noise level. Experimental results on realistic size generators show that the reconstruction of the clock control sequence and consequently the initial state of the clocking subgenerator is feasible up to a relatively high level of noise.

References

1. Barkan, E., Biham, E., Keller, N.: Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication, In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 600–616. Springer, Heidelberg (2003)
2. Chambers, W., Golić, J.: Fast Reconstruction of Clock-Control Sequence. Electronics Letters 38(20), 1174–1175 (2002)
3. Chambers, W.G., Jennings, S.M.: Linear Equivalence of Certain BRM Shift-register Sequences. Electronics Letters 20(24), 1018–1019 (1984)
4. Coppersmith, D., Krawczyk, H., Mansour, Y.: The Shrinking Generator, Advances in Cryptology. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 22–39. Springer, Heidelberg (1994)
5. Eppstein, D.: Finding the k-Shortest Paths. In: Proc. of the 35th IEEE Symposium on Foundations of Computer Science, Santa Fe, pp. 154–165 (1994)
6. Golić, J., Menicocci, R.: Edit Distance Correlation Attack on the Alternating Step Generator. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 499–512. Springer, Heidelberg (1997)
7. Golić, J., Mihaljević, M.: A Generalized Correlation Attack on a Class of Stream Ciphers Based on the Levenshtein Distance. Journal of Cryptology 3(3), 201–212 (1991)
8. Günther, C.G.: Alternating Step Generators Controlled by de Bruijn Sequences. In: Price, W.L., Chaum, D. (eds.) EUROCRYPT 1987. LNCS, vol. 304, pp. 5–14. Springer, Heidelberg (1988)
9. Hirschberg, D.: Serial Computations of Levenshtein Distances. In: Apostolico, A., Galil, Z. (eds.) Pattern Matching Algorithms, pp. 123–141. Oxford University Press, Oxford (1997)

10. Johansson, T.: Reduced Complexity Correlation Attacks on Two Clock-Controlled Generators. In: Ohta, K., Pei, D. (eds.) ASIACRYPT 1998. LNCS, vol. 1514, pp. 342–356. Springer, Heidelberg (1998)
11. Naor, D., Brutlag, D.: On Near-Optimal Alignments of Biological Sequences. J. Comput. Biology 1(4), 349–366 (1994)
12. Oommen, B.: Constrained String Editing. Inform. Sci., 40(9), 267–284 (1986)
13. Sankoff, D., Kruskal, J.B.: Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison. Addison Wesley, Reading (1983)

Solving a Practical Examination Timetabling Problem: A Case Study

Masri Ayob[1], Ariff Md Ab Malik[1], Salwani Abdullah[1], Abdul Razak Hamdan[1], Graham Kendall[2], and Rong Qu[2]

[1] Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia
43600 Bangi,Selangor, Malaysia
{masri,salwani,arh}@ftsm.ukm.my, ariff215@salam.uitm.edu.my
[2] ASAP Research Group, School of Computer Science and Information Technology,
The University of Nottingham,Nottingham NG8 1BB, UK
{gxk,rxq}@cs.nott.ac.uk

Abstract. This paper presents a Greedy-Least Saturation Degree (G-LSD) heuristic (which is an adaptation of the least saturation degree heuristic) to solve a real-world examination timetabling problem at the University Kebangsaan, Malaysia. We utilise a new objective function that was proposed in our previous work to evaluate the quality of the timetable. The objective function considers both timeslots and days in assigning exams to timeslots, where higher priority is given to minimise students having consecutive exams on the same day. The objective also tries to spread exams throughout the examination period. This heuristic has the potential to be used for the benchmark examination datasets (e.g. the Carter datasets) as well as other real world problems. Computational results are presented.

Keywords: Timetabling, Heuristic, Graph colouring.

1 Introduction

The examination timetabling problem is characterised by assigning a set of exams into a limited number of timeslots subject to a set of constraints (see [1], [2], [3], [12], [16], and [20]). These constraints are usually classified as hard and soft constraints. The hard constraints must be completely satisfied where solutions that satisfy all the hard constraints are called *feasible*. On the other hand, there might be some requirements that are not essential but should be satisfied as far as possible, which are referred to as soft constraints. Common hard constraints for examination timetabling problem are: (i) no student should be required to sit two exams at the same time and (ii) the scheduled exams must not exceed the room capacity. However, in practical examination timetabling problem, there are many other constraints, which are vary among institutions. Similarly in our dataset, we have some unique hard constraints, which we describe in section 2.

A particularly common soft constraint aims to spread exams as evenly as possible throughout the schedule. Due to the complexity of the problem, it is not usually possible to have solutions that do not violate some of the soft constraints. Indeed, the

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 611–624, 2007.
© Springer-Verlag Berlin Heidelberg 2007

quality of the timetable is usually evaluated based on some cost which measures the violation of the soft constraints. A weighted penalty value is associated with each violation of a soft constraint and the objective is to minimize the total penalty value.

Various approaches have been developed for solving examination timetabling problems. These include graph-based heuristics [3], tabu search [4], evolutionary algorithms ([5], [6], [13], [14], and [15]), simulated annealing [7], hyper-heuristics [8], and ant algorithms [9]. [8], for example, employed a tabu search approach to search a sequence of graph heuristics in constructing examination timetables within a hyper-heuristic framework. A move in tabu search chooses a new heuristic list by randomly swapping two graph heuristics in the previous heuristic list. The newly visited heuristic lists are added to the tabu list to avoid re-visiting them. To reduce the computational time, [8] introduced a *failed list* that stored any heuristic list that could not generate a feasible solution. Results showed that the approach can produce good quality solutions that are within the range of the best results reported in the literature for both exam and course timetabling problems.

Graph based heuristics were among the earliest approaches to be used for the timetabling problem ([10] and [3]). They are used as constructive heuristics, constructing solutions by ordering the exams that have not yet been scheduled, according to the perceived difficulty in scheduling that exam into a feasible timeslot. The difficulty of an exam can be represented in various ways such as the degree of conflict it has with other exams, the number of student enrollments etc. Some common graph based heuristics are:

1) **Largest degree first.** This heuristic first schedules the exam that has the largest number of conflicts with other exams.
2) **Colour degree.** Exams with a greater number of conflicts with the exams that have already been scheduled have a higher priority of being scheduled next.
3) **Least Saturation degree.** Exams with fewer feasible slots are scheduled as early as possible. The priority of exams (in the unscheduled list) to be scheduled are changed dynamically during the construction of solution.
4) **Largest weighted degree.** Exams with the higher number of students in conflict are scheduled earlier.
5) **Largest enrolment degree.** Exams with greater number of students are scheduled earlier.

Interested readers are referred to [11], [12], [13], [14], [15], [16], [17], [18], and [19] for more information about examination timetabling.

In this paper, we present a Greedy-Least Saturation Degree (G-LSD) heuristic in order to produce good quality solutions to the real-world examination timetabling problem faced by the University Kebangsaan Malaysia (UKM). The heuristic is an adaptation of the least saturation degree heuristic. We employ a hierarchical problem solving approach to generate solutions for UKM. We utilise a new objective function in order to evaluate the quality of a timetable. The objective function considers both timeslots and days in assigning exams to timeslots. Higher priority is given to minimise students having consecutive exams in the same day. The new objective function differs from the standard proximity cost ([20]), where the only measure is how close the exams are, with no account being taken of exams spanning days (i.e. two consecutive exams, on the same day, are penalised the same as an exam in the

evening and an exam the following morning). The new objective function, and the G-LSD heuristic, can be applied to the standard benchmark examination datasets ([20]) by adding a new variable *day* to each timeslot.

2 Problem Definition

Until recently, human schedulers at the Universiti Kebangsaan Malaysia (UKM) were human decision makers who applied a greedy assignment procedure, based on their experience with a little guidance from computer software to avoid clashes. They did not consider students sitting two/three consecutive exams a day. They would like to take into account spreading exams evenly, and fairly, throughout the timetable but the size/complexity of the problem makes this unrealistic. Therefore, the human scheduler only concerns themselves with the constraint of not assigning a student to sit more than one exam in a given timeslot. Even this procedure usually takes the manual schedulers more than two weeks. After circulating the exam timetable to students, the schedulers invariably receive many complaints from students/lecturers. When students complain about sitting three consecutive exams in a day, they are rescheduled, with an emphasis on making as few adjustments to the rest of the timetable as possible. Usually, if this happen (i.e. complaints about three consecutive exams in a day), a new timeslot may be added (normally on Saturday) and the middle exam is scheduled to this new timeslot. This incurs extra overhead costs. Therefore, this work is motivated by attempting to automate this process, as much as possible, as well as take into account more of the requirements. Indeed, the authors based at UKM have been tasked to this do by the management at UKM.

In this paper, we present our approach and experience in solving the examination timetabling problem for Semester I, year 2006 at UKM. The dataset (UKM06-1) has been preprocessed based on the supplied data which contains 818 exams, 14,047 students, 75,857 enrollments, 42 timeslots and 15 exam days (this excludes the weekend break). The UKM06-1 dataset is held in four text files: UKM06-1.stu, UKM06-1.slt, UKM06-1.rom and UKM06-1.isl, which represent student enrollment, slot, room and isolated exams definition, respectively. The files are available at `http://www.ftsm.ukm.my/jabatan/tk/masri/Exam/`. The exam timetabling problems in UKM (particularly) have different datasets for each semester. In most cases, each student registers for different set of courses. That is the exam timetable of each semester is only valid for that semester. Therefore, in practice, the exam timetable need be generated at the end of each semester.

The dataset has 3 weeks examination period. Each week has 5 exam days (Monday to Friday). Each day has 3 timeslots (morning, afternoon and evening), except Friday which has 2 timeslots (morning and evening only). In order to model the real-world timeslots (in days), we present the vector of days in Fig. 1.

(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 8, 8, 8, 9, 9, 9, 10, 10, 10, 11, 11, 11, 12, 12, 15, 15, 15, 16, 16, 16, 17, 17, 17, 18, 18, 18, 19, 19)

Fig. 1. Vector of days

In Fig. 1, we can see that in most cases we have three entries for each day (e.g. day 1, we have three '1') except on Friday i.e. on day 5, 12 and 19; there are only two *5* entries (representing two timeslots on Friday). Saturdays (days 6 and 13) and Sundays (day 7 and 14) are missing because there are no examinations on Saturday and Sunday. A corresponding timeslot vector is presented in Fig. 2.

(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 57)

Fig. 2. Vector of timeslots

Fig. 2 shows that the timeslots are represented as indexes. Timeslots 1, 2 and 3 refer to day 1, timeslots 4, 5 and 6 refer to day 2, etc. Note that on day 5 (Friday the first week) there are only 2 timeslots i.e. 13 and 15 (morning and evening sessions). Since there is no afternoon session, we do not use a timeslot with an index of 14. The reason is that we want to utilise different weights since the students have gaps (2 gaps in this case i.e. 15-13=2) even though the exams are scheduled on the same day. Also notice that the timeslot indexes for Saturday the first week (16, 17 and 18) and Sunday the first week (19, 20, and 21) are missing because there are no exams scheduled on these days. The same representation is used for the second and third weeks of the exam period.

The input for the examination timetabling problem can be stated as follows:

- N is the number of exams;
- E_i is the i^{th} exam where $i \in \{1,....,N\}$;
- e_i is number of students sitting exam E_i where $i \in \{1,....,N\}$;
- B is the set of all N exams, $B=\{E_1,..., E_N\}$;
- D is the number of days;
- T is the given number of timeslots (including missing timeslots);
- M is the number of students;
- t_i specifies the assigned time slot for exam E_i, where $t_i \in \{1,..,T\}$ and $i \in \{1,..,N\}$;
- d_i specifies the assigned day for exam E_i, where $d_i \in \{1,..,D\}$ and $i \in \{1,..,N\}$;
- $C=(c_{ij})_{NxN}$ is the conflict matrix where each element denoted by c_{ij}, $(i,j \in \{1,..,N\})$ is the number of students taking both exams E_i and E_j where $c_{ij}=0$ for $i=j$;
- $\Delta t = |t_i - t_j|$ is the timeslot different between exam E_i and E_j;
- $\Delta d = |d_i - d_j|$ is the day different between exam E_i and E_j;
- β_{ij} is a decision variable where $\beta_{ij}=1$ if $c_{ij}>0$ and $i \neq j$; or 0 otherwise.

The constraints of our examination timetabling problem (excluding constraints for assigning exams to rooms) are:

1) All exams must be scheduled and each exam must be scheduled only once.

$$\sum_{s=1}^{T} \lambda_{is} = 1 \qquad \text{for all } i \in \{1,..,N\}. \tag{1}$$

where

$$\lambda_{is} = \begin{cases} 1 & \text{if exam } E_i \text{ is assigned to timeslot } s; \\ 0 & \text{otherwise;} \end{cases} \quad (2)$$

2) No student can sit two exams concurrently. If exam E_i and E_j are scheduled in slot s, the number of students sitting both exams E_i and E_j must be equal to zero, i.e. $c_{ij}=0$.

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij} . x(t_i, t_j) = 0. \quad (3)$$

where

$$x(t_i, t_j) = \begin{cases} 1 & \text{if } \quad t_i = t_j; \\ 0 & \text{otherwise;} \end{cases} \quad (4)$$

3) For each timeslot t, the number of students sitting for exams ($Students_t$) must not exceed the maximum number of seats ($MaxSeats$) i.e. 2,400 seats per timeslot for the UKM06-1 dataset.

$$Students_t \leq MaxSeats \qquad \text{for } t \in \{1,..,T\}\ ; \quad (5)$$

4) No student can sit 3 consecutive exams in a day.

$$\text{If } c_{ij} \neq 0;\ c_{ik} \neq 0;\ t_i = x;\ i \neq j \neq k;\ [t_j = x+1 \ \textbf{OR}\ t_j = x-1] \text{ and } d_i = d_j; \\ \text{then } d_k \neq d_i; \quad \text{for all } i,j \in \{1,...,N\}; \quad (6)$$

Due to the complexity of the problem, we have partitioned the problem into two sub problems. That is, assigning exams to timeslots (the same as the capacitated examination timetabling problem but with extra hard constraints, i.e. we impose constraint 4) and the room assignment problem. In this paper, we only consider constraints related to the capacitated examination timetabling problem. Constraints for the room assignment problem will be the subject of our future work.

Before assigning exams to timeslots the supplied data requires some preprocessing. Firstly, we exclude courses with no exams and combine exams that have to be scheduled together into a single exam. We also exclude courses from the Law faculty. These only have two timeslots per day, as their exam period is longer than the other exams. We also exclude co-curriculum courses, which have different (shorter) timeslots. In this work, we use a hierarchical problem solving approach, which focuses on the first stage that is assigning exams to timeslots.

The output of the first stage will be used as input to subsequent stages. The next two stages, which assign exams to (suitably sized) rooms and schedules Law exams into slots and rooms, are not considered here and will be the focus of future work. Since the co-curriculum courses are university level courses which have many student

enrollments, from different faculties, and have shorter exam periods, i.e. different timeslots, they are usually scheduled outside the examination weeks. Therefore, scheduling the co-curriculum exams can be solved independently.

To indicate the density of the conflicting exams, we use *Conflict Density* as defined in [20] and many other works. The *Conflict Density* can be calculated as:

$$\textbf{\textit{Conflict Density}} \quad = \quad \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left(\frac{\beta_{ij}}{N}\right)}{N} \tag{7}$$

Higher value of *Conflict Density* shows that, on average many exams are in conflict with each other.

3 Objective Function

As in the standard benchmark datasets, wherever possible, examinations should be spread out over timeslots so that students have large gaps between exams. In order to adhere with practical (real world) issues, we introduce our new cost function, which is adapted from the proximity cost ([20] and the objective function proposed by [13] and [14]). Our *Penalty Cost*, F, is calculated as follows:

$$\textbf{Minimise} \quad F \quad = \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij} . Penalty \quad (t_i, t_j)}{M} \tag{8}$$

where

$$penalty \quad (t_i, t_j) = \begin{cases} 2^{(5-\Delta t)(2-\Delta d)} \\ \\ 0 \end{cases} \tag{9}$$

Equation 9 presents a weighted *penalty* value that reflects the cost of assigning exam E_i and E_j to a timeslot. These being 0, 1, 2, 4, 8, 16, 64 and 256 where the cost is zero if the gap of timeslot for exam E_i and E_j is greater than 5 or the day gap is greater than 2. We only penalise up to a maximum of 5 timeslots in order to adhere to the well established proximity cost proposed by [20]. We also limit the penalty up to a maximum of 2 days because a 2 day gap between examinations gives enough free time for students. This observation is based on a pilot survey we have carried out at UKM.

The new objective function (equations 8 and 9) aims to minimise the number of students having two exams in a row on the same day (highest penalty) and on the next day (lower penalty); and attempts to spread out exams over timeslots (lowest penalty). Indeed, these formulations emphasises avoiding students having two consecutive exams on the same day instead of avoiding having two consecutive exams on different days. The penalty for students having two consecutive exams on the same

day (*penalty=256*) is higher than the penalty value for students having two consecutive exams on different days (*penalty=16*). This factor is not highlighted in the objective function proposed by [13], [14], and [20]. In fact, [20] ignores the day effect by assuming that the practical time gap between each consecutive timeslot is the same. The objective function introduced in [13] penalised consecutive exams on the same day. That is, [13] only minimised the number of students having two consecutive exams on the same day without spreading exams over timeslots. Subsequently, [14] enhanced the objective function that was proposed in [13] by giving a higher penalty (3) to students having two consecutive exams in the same day and a lower penalty (1) for two consecutive exams spread across two consecutive days. By adapting the three objective functions, we proposed a *Penalty Cost* that combines the features of the above functions into one.

4 Heuristic for the Examination Timetabling Problem

This work presents a heuristic procedure which we call Greedy Least Saturation Degree (G-LSD). We use the term "greedy" because our heuristic attempts to assign each exam to the best timeslot which satisfies all the hard constraints. We randomly assign exams to timeslots when the exam has no conflict with the exams that have already been scheduled. While assigning exams to timeslots, we also ensure that all hard constraints are satisfied. This is an adaptation of the least saturation degree heuristic for classical graph colouring problem. Fig. 3 presents the pseudo-code for the G-LSD. The idea is to give a higher priority to exams still to be scheduled that have the least number of available slots, where the priority changes dynamically during the scheduling process.

In the initialisation step, all exams in B are reset. That is, the number of available timeslots for each exam is set to the maximum available slot and the exam's status is changed to unscheduled and copied into the unscheduled exam set B'={E_1', E_2'….,E_N'}. The heuristic first arranges the unscheduled exams in B′ in non-decreasing order of the number of available timeslots, then in non-increasing order of the number of conflicts they have with other exams (in B) and, finally, by non-increasing order of the number of student enrollments. The heuristic chooses the first exam in B', E_i', and assigns it to the best timeslot that minimises F (equation 8) and subject to the total number of students assigned to the timeslot that does not exceed the maximum seat capacity (i.e. 2400 for UKM06-1 dataset). However, when all slots are available for E_i' (i.e. the exam has no conflict with the exams that have already been scheduled), we randomly choose a timeslot for E_i'. The idea is to allocate the best timeslot for E_i', so as to obtain different solution for each run. While assigning exams to timeslots, we also ensure a clash free schedule (i.e. constraints 1 to 3 are satisfied) and larger exam (student enrollment>=400) are assigned to earlier timeslots (the first two weeks if possible). This is done as, based on discussions with UKM registry officers, they usually assign larger exams to earlier timeslots in order to give longer time for marking larger exams. After assigning exam E_i' to the timeslot, we update the appropriate exam details such as timeslot index, number of available timeslots etc. in B and reduce the number of available timeslots for exams in B' accordingly. Then, we eliminate E_i' from B' and repeat step 2.1 to 2.6 until all the exams have been scheduled, or until E_i' cannot be assigned to any available timeslot.

```
Step 1: Initialisaon
         Initialise all exams in B.
         Copy all exams in B into unscheduled exam set,
         B'={E1', E2'….,EN'}.

Step 2: Assigning Exam to Slot.
         While(B'≠∅)
         {
             2.1   Arrange the exam in B' using Procedure A.
             2.2   Choose the first exam, Ei' in B'.
             2.3   If all timeslots are available for Ei';
                     Then, Randomly choose available timeslot, ti,
                     s.t. the limit of seat capacity of the
                     timeslot and larger size exam should be
                     assigned to earlier timeslot.
             2.4   Else, choose timeslot, ti which has the minimum
                     F, s.t. the limit of seat capacity of the
                     timeslot and larger size exam should be
                     assigned to earlier timeslot.
                     If there is no clash free timeslot for Ei',
                     exit while loop and repeat step 1.
             2.5   Update appropriate exams in B and reduce the
                     number of available timeslots for exams in B'
                     accordingly.
             2.6   Remove Ei' from B'.
         }end while.

Step 3: Verification
         If the solution is feasible concerning constraint (4),
         return the solution.
         Otherwise, repeat Step 1 to Step 3 until the feasible
         solution is obtained or exceed the time limit.
```

Note: Procedure A arrange unscheduled exams in B' in non-decreasing order of number of available timeslots then with non-increasing order of number of conflicts they have with other exams (in B) and non-increasing order of number of students enrollment.

Fig. 3. Psuedo-code of Greedy-Least Saturation Degree (G-LSD) heuristic

If this occurs, we stop the process and start again (steps 1 to 3). After assigning all exams to timeslots, we verify that all hard constraints (constraint 1 to 4) are satisfied. If the solution is feasible, we end the process. Otherwise, we begin again (steps 1 to 3).

Before assigning exam E_i' to a timeslot we ensure that constraints 1 to 3 are satisfied. However, we can only verify if constraint 4 is satisfied when all exams have been assigned to timeslots. This is because we want to reduce the computational time due to the size of problem and the complexity of computing the three consecutive exams in a day. Moreover, our new objective function is able to avoid assigning students sitting three consecutive exams in a day by giving a very high penalty (i.e. *penalty=256*256*) for this case. Therefore, in many cases, our algorithm produces solutions where no students are sitting three consecutive exams in a day. However, the solution is rejected (infeasible) if there are students sitting three consecutive

exams in a day. Since this is a constructive heuristic, we stop the process when we obtain a feasible solution. Of course, we could extend this procedure to produce many feasible solutions and return the best solution found by repeating step 1 to 3 for a given number of iterations.

5 Results

Our heuristic was implemented using Microsoft Visual C++ 6.0. We ran the experiments on an Athlon 1.47 GHz processor, with 512 MB RAM and Windows XP Professional. The G-LSD heuristic was tested on the real world UKM06-1 dataset and also on the benchmark Carter datasets. In order to transform the Carter datasets into a form that is suitable for use with the recently proposed objective function, we apply a day and timeslots vectors (as in UKM06-1; see Fig. 1 and Fig. 2), except that for benchmark Carter datasets we have three timeslots on Friday and one timeslot on Saturday (as suggested in [14] and [15]). We also introduce a six timeslot gap between Saturday morning and Monday morning to cater for the weekend break. For each test, we perform 100 runs. Since there is a random element in selecting each timeslot, when all timeslots are available, we can usually obtain different solutions for each run. In all tests, we use the new *Penalty Cost* (see section 3) function to evaluate the quality of timetable.

Table 1 shows the characteristics of original UKM06-1 and Carter datasets. For capacitated examination timetabling problems, no justification was stated on how the seat capacity limits were set (see [13], [14], and [15]). For example, the seat capacity limit for Car-s-91 was set to 1550 seats. If we divide the number of enrollment by the seat capacity, we need at least 37 timeslots (in ideal case where we can easily assign exams to any slots, i.e. exams have no conflict with each other) such that all students can have a seat in the exam room. We assume this is why [13] increased the number of timeslots from 35 to 51 for Car-s-91 and 23 timeslots to 35 for Tre-s-92. This work also introduces the sta-f-83 instance as a capacitated problem. Later (test B) we run a series of experiments to establish the number of timeslots and the seating capacity required for this revised dataset instance.

Table 1. Characteristics of benchmark and UKM06-1 examination problems

Dataset	Exams	Times-lots	Days	Seat capacity	Students	Enrollment	Conflict density
UKM06-1	818	42	15	2400	14047	75857	0.05
Car-f-92	543	32	12	2000	18419	55522	0.14
Car-s-91	682	35/51	13/19	1550	16925	56877	0.13
Kfu-s-93	461	20	8	1955	5349	25113	0.06
Tre-s-92	261	23/35	9/13	655	4360	14901	0.18
Sta-f-83	139	13	5	-	611	5751	0.14

*Note: * Previously, days have not been properly defined. Based on discussion in [14] and [15], we compute the number of examination days (excluding day off).*
***Seat capacity was introduced later by [13] for five Carter datasets.*

For UKM06-1, we ran an initial test (we call it test A) to establish the best number of timeslots for this dataset. We varied the number of timeslots whilst fixing the

maximum room capacity (i.e. 2,400 seats as currently used). In practice, the number of timeslots available is 42. Actually, due to the difficulties of scheduling large exams, manual schedulers usually allocate certain timeslots for some exams. For example, exams ZZZT1032 and ZZZT1042 which have 2146 and 2110 student enrollments, respectively, were allocated to one entire timeslot for each exam. As a result no other exams were scheduled in these timeslots. Indeed, these exams were enrolled by many students from various faculties, i.e. the conflict density for these exams is very high. However, by using our approach, these exams can be scheduled together with other exams. Although, by setting the number of timeslots is equal to 42 and the maximum room capacity is equal to 2400, human schedulers still struggle to provide good quality timetables. Indeed, they have no clear idea what is the criterion for a good quality timetable and how they can manage to find one, even manually. They always receive many complains after the tentative exam timetable has been circulated to students/lecturers. Therefore, we design test A to show that they can reduce the number of timeslot (that will ultimately reduce the management cost) without scarifying too much on solution quality. Results of this test, is shown in Table 2.

Table 2. Results for the test A on UKM06-1 when varying the number of timeslots

Time-slots	Exam Days	Seat capacity	Number of clash free solutions	Feasible solutions (%)	Average feasible solution, F_{avg}	Best feasible solution, F_{min}
42	15	2400	100	79	4.76	4.25
41	15	2400	100	70	5.30	4.81
40	14	2400	100	35	6.08	5.44
40	14	2106	0	-	-	-
40	14	2250	100	36	6.21	5.86

In Table 2, *'Number of clash free solutions'* indicates the total number of solutions obtained (ignoring violations of students sitting three consecutive exams in a day) over the 100 runs; '*Feasible solutions*' indicates the number of solutions obtained that satisfy all the hard constraints (1 to 4), including no student sitting three consecutive exams in a day. Results in Table 2 shows that we obtained 79%, 70% and 35% feasible solutions when seat capacity=2400 and timeslots is equal to 42, 41 or 40 respectively.

Some secondary rooms (i.e. LobiA(DECTAR) and LobiB(DECTAR)) allocated to the UKM06-1 dataset are actually not preferable due to practical constraints, so we eliminate these rooms from the list of available rooms. For UKM dataset, we also eliminate PSeni(DECTAR) from the list because the room is reserved for Law examinations. Therefore, the total room capacity is reduced to 2340 seats. As a result, we set 2106 or 2250 as the maximum seat capacity limit, i.e. 90% or 96% of the total seat capacity respectively. Table 2 shows that, out of 100 runs, no feasible solution is obtained when the number of timeslots is equal to 40 and seating capacity is equal to 2106. Whereas, we obtained 36 feasible solutions with the best *Penalty Cost,* F_{min} = 5.86 when we used timeslots is equal to 40 and seat capacity is equal to 2250. These results show that our G-LSD is capable of reducing the number of timeslots and avoiding students sitting three consecutive exams a day, as well as spreading out

exams over timeslots. It shows that we can save two timeslots compared to manual scheduler (this might be of interest to the administrator because they could reduce their administration cost). Since the number of timeslots was reduced, the UKM06-1 dataset is now even more constrained.

We carried out a further test (test B) to find out a reasonable number of timeslots required in order to generate feasible solutions (that satisfies all constraints 1 to 4) for the sta-f-83 problem. Initially we set the seat capacity to 2400 such that all students (clash free schedule) have a seat in the exam room. We then, reduced the maximum seating capacity limit to a reasonable value. We also varied the number of timeslots for sta-f-83 starting from 13 timeslots until we found feasible solutions (20 timeslots). Table 3 shows the results of this test. We also extended our experiment to Kfu-s-93, Car-f-92, Car-s-91 and Tre-s-92. We initially tried not changing the number of timeslots and the maximum seat capacity that were set by [20], [13], and [14]. As we are enforcing an additional hard constraint (i.e. avoiding students sitting three consecutive exams in a day), we thought it is worth carrying out some further analysis on these datasets. Table 3 also shows the results of the experiments.

Table 3. Results of test B on Sta-f-83 when varying the number of timeslots

Dataset	Time-slots	Exam Days	Seat capacity	Number of clash free solutions	Feasible solutions	Average feasible solution, F_{avg}	Best feasible solution, F_{min}
Sta-f-83	19	7	2400	100($\mathbf{F'_{min}}$=241.31)	0	-	-
Sta-f-83	20	8	2400	100($\mathbf{F'_{min}}$=189.13)	46	215.58	189.13
Sta-f-83	20	8	428	100($\mathbf{F'_{min}}$=224.41)	18	258.05	241.52
Kfu-s-93	20	8	1955	54($\mathbf{F'_{min}}$**=57.37**)	0	-	-
Kfu-s-93	20	8	2400	67($\mathbf{F'_{min}}$**=55.48**)	0	-	-
Kfu-s-93	40	15	2250	100($\mathbf{F'_{min}}$**=5.86**)	36	6.21	5.86
Car-f-92	49	19	2400	100($\mathbf{F'_{min}}$**=3.02**)	0	-	-
Car-f-92	51	19	2400	100($\mathbf{F'_{min}}$**=2.76**)	10	2.87	2.76
Car-s-91	51	19	2400	100($\mathbf{F'_{min}}$**=4.11**)	0	-	-
Car-s-91	53	20	2400	100($\mathbf{F'_{min}}$**=3.54**)	16	3.98	3.84
Tre-s-92	35	13	655	100($\mathbf{F'_{min}}$**=8.18**)	11	8.88	8.55

Note: The exams day excluded day off (i.e. weekend breaks or public holidays)

Table 4. New characteristics of existing benchmark and UKM06-1 examination datasets

Dataset	Exams	Timeslots	Days	Seat capacity	Students	Enrollment	Conflict density
UKM06-1	818	40	14	2250	14047	75857	0.05
Car-f-92	543	51	19	2400	18419	55522	0.14
Car-s-91	682	53	20	2400	16925	56877	0.13
Kfu-s-93	461	40	15	2250	5349	25113	0.06
Tre-s-92	261	35	13	655	4360	14901	0.18
Sta-f-83	139	20	8	428	611	5751	0.14

Note: We set the maximum seat capacity for sta-f-83 as 428 i.e. 70% of the sum of students.

As a result of the experiments reported above, we propose revised characteristics of the capacitated examination datasets (see Table 4) with the added hard constraint, new *Penalty Cost* and day vector.

6 Conclusion and Future Work

This paper has presented our experience in solving a real-world examination timetabling problem at the University Kebangsaan Malaysia (UKM) with an objective to minimise the *Penalty Cost* using a new objective function. The new objective function attempts to minimise the number of students having two consecutive exams on the same day (highest penalty) and on the next day (lower penalty). It also attempts to spread out exams over timeslots (lowest penalty). Due to the complexity of the problem, we divided the problem into two sub problems; i.e. capacitated examination timetabling and room assignment problems. We employed a hierarchical problem solving approach that has three stages to solve the problem. This work focused on the first stage, that is to assign exams to timeslots subject to a clash free schedule, no student sitting three consecutive exams in a day and total students sitting exams per slot should not exceed the seating capacity. The output of the first stage serves as input for the subsequent stages. The next two stages, which assign exams to rooms and scheduled law exams to slots and rooms will be discussed in our future work.

The Greedy Least Saturation Degree (G-LSD), presented in this work, is an adaptation of basic least saturation degree heuristic for the classical graph colouring problem. As in the standard least saturation degree heuristic, the idea is to give high priority to the exams that are still to be scheduled and which have the least available slots, where the priority changes dynamically during the scheduling process. G-LSD was tested on the UKM06-1 dataset using the new *Penalty Cost* as an evaluation function. Since this is a new *Penalty Cost*, we are not able to make a comparative study. Moreover, no previous work has attempted to impose the hard constraint of no student sitting three consecutive exams in a day. Indeed, it is not fair to compare our solution with manually generated solution since their solution was generated without considering three consecutive exams in a day and not attempted to spread exams over timeslot (i.e. they do not have an objective function).

In order to adhere with practical (real world) constraints, and to further test our approach, we add a new hard constraint to the Carter benchmark exam timetabling dataset in [20], i.e. no student should sit three consecutive exams in a day. That is, we proposed revised characteristics of the capacitated examination datasets with the added hard constraint, new *Penalty Cost* and day vector. This can further validate the approach we have suggested here.

Since this approach has been adapted from a graph colouring approach, it could be applied to other wide range of problems such as course timetabling and school timetabling which can be modeled as graph colouring problems.

We are currently designing and implementing a constructive heuristic for assigning exams to rooms. In order to solve the room assignment problem, we have to model the room assignment problem and design a new objective function because currently, no previous researchers have defined the objective function for room assignment. Our future work will also concentrate on scheduling examinations for a Law faculty which is slightly different from other faculties at the University Kebangsaan Malaysia. This

faculty only has two slots per day (because the exam periods are longer than other exams i.e. they start at 8:30am and 2:30pm). We will also try to schedule invigilators (which is carried manually at the moment) and to incorporate this requirement into the scheduling system.

Acknowledgment. This work was supported by The University Kebangsaan Malaysia (UKM). We wish to thank Nor Ashikin Baharom from Computing Center, UKM and Student Registry Department for fruitful discussion and contribution of UKM dataset.

References

1. McCollum, B.: University timetabling: Bridging the gap between research and practice. In: Burke, E.K., Rudová, H. (eds.): Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling. 30th August-1st September 2006, Brno, Czech Republic, pp. 15–35 (2006)
2. Burke, E.K., Elliman, D.G., Ford, P.H., Weare, R.F.: Examination timetabling in British universities - A survey. In: Burke, E.K., Ross, P. (eds.) Practice and Theory of Automated Timetabling. LNCS, vol. 1153, pp. 76–92. Springer, Heidelberg (1996)
3. Burke, E.K., Kingston, J., de Werra, D.: In: Gross, J., Yellen, J. (eds.) Applications to timetabling. Handbook of Graph Theory, pp. 445–474. Chapman Hall/CRC Press (2004)
4. Di Gaspero, L., Schaerf, A.: Tabu search techniques for examination timetabling. In: Burke, E., Erben, W. (eds.) PATAT 2000. LNCS, vol. 2079, pp. 104–117. Springer, Heidelberg (2001)
5. Erben, W.: A grouping genetic algorithm for graph coloring and exam timetabling. In: Burke, E., Erben, W. (eds.) PATAT 2000. LNCS, vol. 2079, pp. 132–158. Springer, Heidelberg (2001)
6. Côté, P.: A hybrid multi-objective evolutionary algorithm for the uncapacitated exam proximity problem. In: Burke, E.K., Trick, M.A. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 294–312. Springer, Heidelberg (2005)
7. Dowsland, K.: Off the peg or made to measure. In: Burke, E.K., Carter, M. (eds.) PATAT 1997. LNCS, vol. 1408, pp. 37–52. Springer, Heidelberg (1998)
8. Burke, E.K., McCollum, B., Meisels, A., Petrovic, S., Qu, R.: A graph-based hyper-heuristic for educational timetabling problems. European Journal of Operational Research 176, 177–192 (2007)
9. Eley, M.: Ant algorithm for the exam timetabling problem. In: Burke, E.K., Rudová, E.K. (eds.) Practice and Theory of Automated Timetabling. LNCS, vol. 1153, pp. 167–180. Springer, Heidelberg (1996)
10. Welsh, D.J.A., Powell, M.B.: The upper bound for the chromatic number of a graph and its application to timetabling problems. The Computer Journal 11, 41–47 (1967)
11. Abdullah, S., Ahmadi, S., Burke, E.K., Dror, M.: Investigating Ahuja-Orlin's Large Neighbourhood Search Approach for Examination Timetabling. OR Spectrum 29(2), 351–372 (2007)
12. Qu, R, Burke, E., McCollum, B., Merlot, L.T.G., Lee, S.Y.: A survey of search methodologies and automated approaches for examination timetabling. Technical Report No. NOTTCS-TR-2006-4, School of Computer Science & IT, University of Nottingham (2006)

13. Burke, E.K., Newall, J.P., Weare, R.F.: A Memetic Algorithm for University Exam Timetabling. In: Burke, E.K., Ross, P. (eds.) Practice and Theory of Automated Timetabling I. LNCS, vol. 1153, pp. 3–21. Springer, Heidelberg (1996)
14. Burke, E.K., Newall, J.P., Weare, R.F.: Initialization strategies and diversity in evolutionary timetabling. Evolutionary Computation 6(1), 81–103 (1998)
15. Burke, E.K., Newall, J.P.: A multistage evolutionary algorithm for the timetable problem. IEEE Trans. Evolutionary Computation. 3, 63–74 (1999)
16. Carter, M.W.: A survey of practical applications of examination timetabling. Operations Research 34, 193–202 (1986)
17. Merlot, L.T.G., Boland, N., Hughes, B.D., Stuckey, P.J.: A hybrid algorithm for the examination timetabling problem. In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. LNCS, vol. 2740, pp. 207–231. Springer, Heidelberg (2003)
18. Asmuni, H., Burke, E.K., Garibaldi, J.M., McCollum, B.: Fuzzy multiple ordering criteria for examination timetabling. In: Burke, E.K., Trick, M.A. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 334–353. Springer, Heidelberg (2005)
19. Ayob, M., Burke, E.K., Kendall, G.: An iterative re-start variable neighbourhood search for the examination timetabling problem. In: Burke, E.K., Ross, P. (eds.) Practice and Theory of Automated Timetabling. LNCS, vol. 1153, pp. 336–344. Springer, Heidelberg (1996)
20. Carter, M.W., Laporte, G., Lee, S.Y.: Examination timetabling: Algorithmic strategies and applications. Journal of Operational Research Society 47(3), 373–383 (1996)

A Geometric Design of Zone-Picking in a Distribution Warehouse

Ying-Chin Ho[1], Hui Ming Wee[2], and Hsiao Ching Chen[2,*]

[1] Institute of Industrial Management
National Central University
Chung-Li, 32045, Taiwan
[2] Department of Industrial Engineering
Chung Yuan Christian University,
Chungli, 32023, Taiwan, ROC
Fax: 886-3-2654499
sayoho@yahoo.com.tw

Abstract. In this study, we considered a regular logistic warehouse into various geometric zone-picking types. Eight zone planning methods, six similarity coefficients, two order-batching methods and one clustering algorithm are used to improve the warehouse system. In all, there are 96 (8x6x2 x1) combinations of these methods. Each combination is tested with thirty randomly generated test problems. The total order-picking travel distances are collected for performance evaluation. Furthermore, we solved the storage-location assignment problem of each test problem by dividing the logistic warehouse into various zone types, so that we can see how much improvement could be achieved by converting a regular warehouse into different geometric zone-picking warehouses.

Keywords: Distribution center, geometric Zone-Picking, Similarity coefficient, SA(Simulated Annealing).

1 Introduction

The role of distribution centers is becoming more and more important to the logistic industry. There is much pressure on suppliers delivering goods to customers in an efficient way. Modern distribution centers (DC) provide more functions in the business environment. For example, customers can order some specified merchandise in stores, by telephone or internet and get them in 24 hours. In order to provide this service successfully, companies need distribution centers with high efficient order-picking operations. Researchers have shown that order-picking operations account for about 65% of the total operating cost of a warehouse [2] and their travel time accounts for roughly 50% of all order-picking operations [24]. To optimize the efficiency of a distribution warehouse, we need to obtain the optimization order-picking operations. Researchers have studied different problems and proposed many methods for them to achieve this goal. For example, many researchers have studied the route-planning problem to order-picking. Hall [9] develops distance approximation methods for five order-picking strategies, which include traversal strategy, midpoint strategy,

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 625–636, 2007.
© Springer-Verlag Berlin Heidelberg 2007

largest-gap strategy, optimal routing, and minimal strategy. Petersen [16] evaluates the performance of five heuristic routing heuristics in comparison with the optimal method proposed by Ratiff and Rosenthal [18]. He finded that the best solutions found by heuristics are on average 10% over the optimal solutions. De Koster and Van Der Poort [3] also conduct a comparison study between the optimal method and the heuristic routing methods. Roodbergen and De Koster [20] investigate the problem of route-planning problems in a warehouse with multiple cross aisles. They provided several heuristics to determine order-picking routes.

Some studies focus on the order-batching problem. Researchers have shown that order batching can reduce the order-picking travel distance if orders with similar picking locations can be picked together. Armstrong *et al.* [1] investigate the order-bathing problem with fixed batch sizes and present an integer programming model for it. Elasyed [5] propose four algorithms that batch orders to minimize the total distance traveled by the S/R (Storage/Retrieval) machine in an AS/RS (Automated Storage/Retrieval System). Other studies that also focus on the order-batching problem in an AS/RS include Elsayed and Stern [6], Hwang *et al.* [13], and Hwang and Lee [14]. Gibson and Sharp [7] adopt computer simulations to compare two order-batching procedures. Various factors, e.g., travel metric, warehouse representation, item location assignments, number of items per order and the total number of orders, are considered in their study. Rosenwein [21] compares various heuristics for order batching by using various metrics to approximate the 'closeness' between a pair of orders and provides a quantifiable basis for assigning orders to batches. De Koster *et al.* [4] compare two groups of order-batching algorithms – the Seed algorithms (CPU time saving) and Time Savings algorithms (CPU time consuming).

There is also a group of studies focusing on the storage-location assignment problem that concerns allocating items to storage locations. Different policies have been proposed for this problem. Hausman *et al.* [10] present three storage-location assignment policies – randomized storage, class-based storage and dedicated storage for automatic warehousing systems. Some location assignment policies are volume-based. In volume-based storage policies, one assigns items to storage locations based on their demand volume. High demand-volume items are placed close to the I/O (Input/Output) point. Schwarz *et al*. [23] conducted simulation experiments to examine the performance of an automatic warehouse with random and volume-based storage. Gibson and Sharp [7] and Gray *et al.* [8] find that locating high-volume products close to the I/O point result in the better efficiency of picking operations. Similar claims can also be found in Coyle *at al.* [2]. Y. C. Ho,, and S.P. Chien., proposed different zone-visitation sequencing strategy, and found that dynamic strategy could be a better strategy. Petersen and Schmenner [17] present four volume-based storage policies and study their performance under different locations of the I/O point. They find the within-aisle policy coupled with the middle I/O location has the best performance in minimizing the order-picking route distance.

This paper will focus on a different problem, i.e., the problem of converting a regular warehouse into various types of geometric zone-picking warehouse. This conversion problem has not received as much attention as the aforementioned problems from researchers. In the experiments, we considered different order-batching methods (since we assumed order batching is adopted in the warehouse) and different route-planning methods. The remainder of this paper is organized as follows. In section 2,

the problem environment and assumptions of this paper are presented. In section 3, the structure of the proposed solution methodology is introduced. In section 3, the order-batching methods and the route-planning methods are considered in the experiments. In section 4, experiments are conducted to test the performance of the methods. In section 5, the analysis of the experimental results are conducted and discussed. Finally, in section 6 summary and conclusion are obtained.

2 Problem Description and Design Methods

The problem environment is a distribution center's warehouse of Taiwan logistic industry. In the warehouse, human pickers perform the picking operations. The warehouse has many parallel picking aisles and two cross aisles – one front cross aisle and one back cross aisle. The I/O point of the warehouse is at one of its corners. The I/O point is the place where pickers receive instructions from the computer for their next picking trips. Figure 1.a gives the example of the warehouse's layout, which is also used in our experiments. In this study, we converted this warehouse into various types of geometric zone-picking warehouses. Figure 1.b shows the different types of the zone-picking warehouse with 2,3,4 equal zones. The 2a,2b denote the two equal zones split the warehouse relative to the length and with. The 3a,3b,3c denote the three

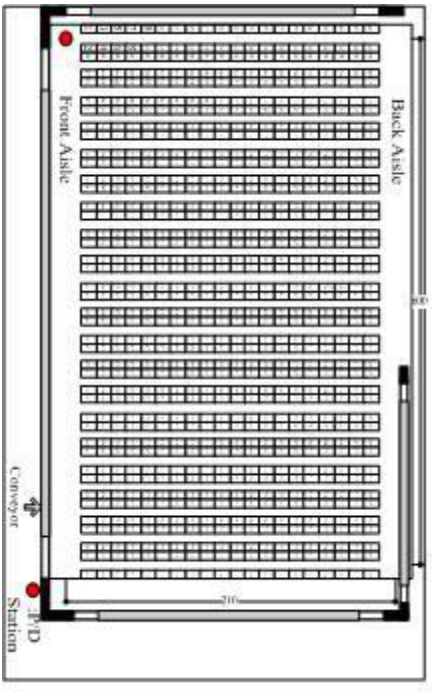


Fig. 1. (a) The layout of an exemplary warehouse

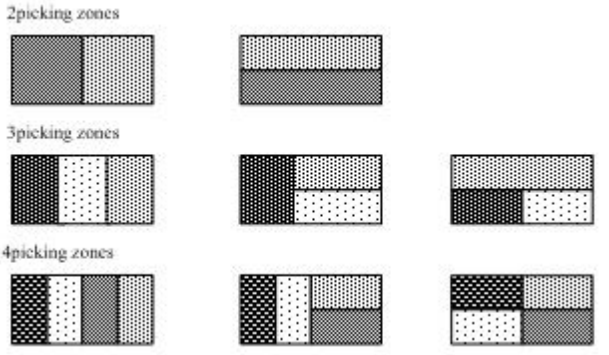


Fig. 1. (b) the layout of the warehouse with 2,3,4 geometric picking zones

equal zones split relative to the length and both length and width. And the 4a,4b,4c denote the four equal zones split relative to the length and width. (see Fig 1.b)) In the zone-picking warehouse, we split the whole order into 2,3,4 sub-orders if its items are distributed in the different zones. Furthermore, each zone has its own I/O point and a conveyor is added to transport boxes of picked items to the same area, where the sub-orders from the same order can be merged together. We assumed that many order-picking shifts are performed in the warehouse. Before an order-picking shift begins, the set of orders that need to be picked in the shift is already known. We refereed to this set of orders as an order pool. In addition, the content of an order pool cannot be changed while the order-picking operations have begun.

3 The Proposed Methods

In this study, the first stage is to determine the items of each zone, while the second stage is to determine the storage locations of all items in each zone. The methods proposed for these two stages are presented in the following sections.

3.1 Methods for Determining the Items in Each Zone

There are three phases in the proposed methods. First, we used a similarity coefficient to measure the similarity coefficients between the order items. Then we used a clustering algorithm to cluster similar items into item groups. The number of item groups must be equal to the number of zones. Finally, we assigned an item group into the different zone area. The six similarity coefficients and one clustering algorithms studied in this paper are presented in the following sections.

3.2 Similarity Coefficients

Similarity coefficients have been widely used in Group Technology. In this paper, six similarity coefficients were collected to measure the similarity between different

items. These coefficients have been used in GT applications to measure the similarities between parts. We used these coefficients to measure two items' similarity based on the orders needing level. These similarity coefficients are presented as follows.

- *Baroni-Urbani and Buser* similarity coefficient (SC-1)

According to Sarker [22], *Baroni-Urbani and Buser* similarity coefficient appears to have advantage than other similarity coefficients, that is, the distribution of its values is more normal (bell shaped) and continuous. The *Baroni-Urbani and Buser* similarity coefficient (BSC_{ij}) between two items i and j is as follows and its value range is [0, 1].

$$BSC_{ij} = \frac{a + (ad)^{1/2}}{a + b + c + (ad)^{1/2}} \tag{1}$$

where,

a: number of orders that need items i and j.
b: number of orders that need item i only.
c: number of orders that need item j only, and
d: number of orders that need neither i nor j.

- *Ochiai* similarity coefficient (SC-2)

According to Romesburg [19], the *Ochiai* similarity coefficient (OSC_{ij}) between two items i and j is defined as follows. Its value rang is also [0, 1].

$$OSC_{ij} = \frac{a}{[(a+b)(a+c)]^{1/2}} \tag{2}$$

where a, b, c, and d are identical to those in Eq. (1).

- *Sorenson* similarity coefficient (SC-3)

The *Sorenson* similarity coefficient (SSC_{ij}) between two items i and j is defined as follows [19] and its value range is also [0, 1].

$$SSC_{ij} = \frac{2a}{2a + b + c} \tag{3}$$

where a, b, c, and d are identical to those in Eq. (1).

- *Jaccard* similarity coefficient (SC-4)

Jaccard similarity coefficient has been widely used in many areas, e.g., numerical taxonomy and GT. The *Jaccard* similarity coefficient (JSC_{ij}) between two items i and j is defined as follows and its value range is [0, 1].

$$JSC_{ij} = \frac{a}{a + b + c} \tag{4}$$

where, where a, b, c, and d are identical to those in Eq. (1).

- Rogers similarity coefficient (SC-5)

The Rogers similarity coefficient (PSC_{ij}) between two items i and j is defined as follows [19]. Rogers similarity coefficient has a value range of [0, 1].

$$RSC_{ij} = \frac{a+d}{[a+2(b+c)c+d]} \tag{5}$$

where a, b, c, and d are identical to those in Eq. (1).

- *Sokal and Sneath* similarity coefficient (SC-6)

Sokal and Sneath [11] similarity coefficient (HSC_{ij}) between two items i and j is defined as follows [11]. The value range of *Sokal and Sneath* similarity coefficient is also [0, 1].

$$SokSC_{ij} = \frac{2(a+d)}{2(a+d)+b+c} \tag{6}$$

where a, b, c, and d are identical to those in Eq. (1).

3.2.1 Clustering Algorithms

According to Figure 2, a clustering algorithm is used to cluster items into item groups. The clustering algorithm is explained as follows.

- Cumulative Clustering Algorithm

CA is a cumulative seed-based clustering algorithm. When CA is adding an item to an item group, it is based on how similar this item is to the entire item group. Detailed steps of CA-1 are as follows.

1. From the item set *IS*, select *NZ* seed items whose sum of their similarity coefficients is the smallest. *NZ* is the number of zones in the warehouse. It is also the number of item groups that will be generated.
2. Every seed item is assigned to an item group, i.e., a seed item is the first item assigned to an item group. Let *IGS* stand for the set of item groups. Delete these selected seed items from *IS*.
3. For every item i in *IS*, the similarity coefficient with every item group would be calculated, G, in *IGS* using the following equations.

$$IGSC_{iG} = \sum_{j \in G} \frac{NofOders(j)}{TotalOrder(G)} \bullet SC_{ij} \tag{7}$$

$$TotalOrder(G) = \sum_{k \in G} NofOrder(k) \tag{8}$$

where,

$IGSC_{iG}$: the similarity coefficient between item i and item group G.
$NofOrder(j)$: the total number of orders that need item j.
$TotalOrder(G)$: the total number of orders that need the items in G.

4. Set *TempSet* =

5. Select an item p (where $p \in IS / TempSet$) whose *IGSC* value with an item group R (where $R \in IGS$) is the greatest, i.e.,
   $IGSC_{pR} = \max\{IGSC_{iG} | i \in IS / TempSet, G \in IGS\}$.
6. Check if adding p to R will cause the number of items in R to exceed *MaxNofItemsAllowed*(R) – the maximum number of items that are allowed in R. If no, add p to R, delete p from *IS*, and proceed to step 7. If yes, add p to *TempSet* and return to step 5.
7. Repeat step 3 to step 6 until all items have been assigned to the item groups.

3.3 Order-Batching Methods

The order-batching methods adopted here are made up of a seed-order selection rule and an accompanying-order selection rules. The flow chart of these order-batching methods is shown in Figure 2. In this section, two seed-order selection rules and one accompanying selection rule are presented. Ho and Tseng [12] have shown that these rules perform well in minimizing the total order-picking distance.

Seed-Order Selection Rule

- Smallest Number of Visited Locations (SNVL) Rule

With SNVL rule, the order visiting the smallest number of picking locations is selected as the seed order.

- Smallest Number of Visited Aisles (SNVA) Rule

With SNPA rule, the order visiting the smallest number of picking aisles is selected as the seed order.

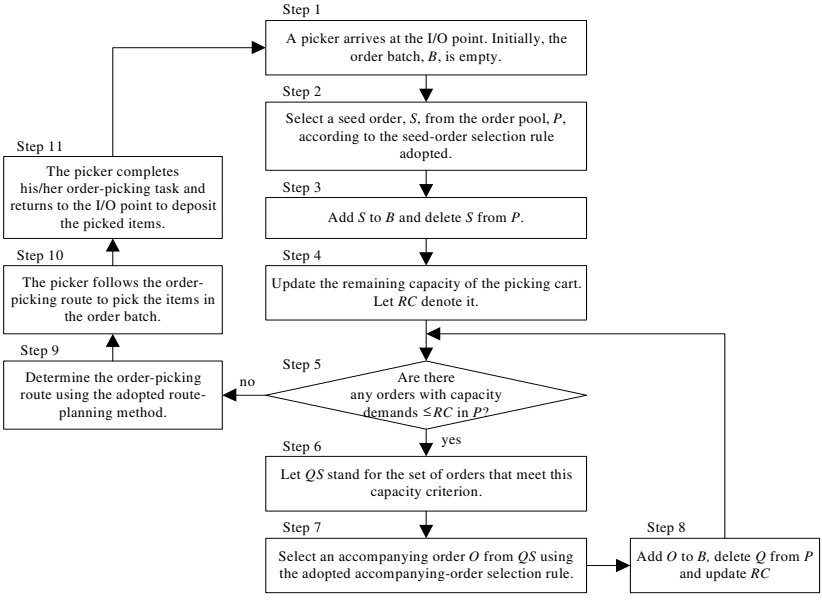


Fig. 2. The flow chart of the adopted order batching methods [12]

3.4 Route Planning Methods

After decided the order-batching methods, we also need order-picking route planning methods to design the route of each order-picking trip. In this study the Largest Gap with Simulated Annealing improvement (LG+SA) method is adopted and explained as follows.

Researchers [16, 17] have shown that LG method outperforms other route-planning methods (e.g., Traversal Strategy, Return Strategy, and Midpoint Strategy) in minimizing the travel distance of an order-picking route.

Ho and Tseng [12] combine LG method and Simulated Annealing (SA) to find the route for a picking trip. They use the route found by LG method as the initial route (solution) for SA. They then use SA to improve the initial route (solution) to find a better solution. Ho and Tseng [12] have shown LG+SA method outperforms LG in minimizing the travel distance of an order-picking route.

4 Experiments

In the study, experiments are conducted to evaluate the performance of the similarity coefficients, the clustering algorithms rules, the seed-order selection rules, and the route-planning methods. The warehouse for our experiments is shown in Figure 1.a. There are twenty storage locations on each side of the aisle in the warehouse. Totally there are 840 storage locations in the warehouse. In the experiments, we divided the warehouse into 2,3,4 equal zones with eight different geometric lay-outs(2b,2b,3a,3b,3c,4a,4b,4c), i.e., each zone will contain 420,280,210 storage locations (see Figure 1.b). The performance measure for the experiments is Total order-picking Travel Distance (TTD). Thirty order pools were randomly generated and there are 300 orders in each picking task. Table 1 summarizes the factors consid ered in the experiments and their levels. Full-factorial experiments are concerned. Totally there are 96 (8x6x1x2x1) combinations of factors. Since each combination is experimented thirty times (using thirty different order pools), totally 2,880 experiments will be conducted.

Table 1. A summary of factors considered in the experiments and their levels

	Factor				
Level	**Zone Picking Type**	**Similarity Coefficient**	**Clustering Algorithm**	**Seed-Order Selection Rule**	**Route-Planning Method**
1	2a	Baroni-Urbani and Buser	Cumulative Clustering Algorithm	SNVL	LG+SA
2	2b	Ochiai		SNVA	
3	3a	Sorenson			
4	3b	Jaccard			
5	3c	Rogers and Tanimoto(1960)			
6	4a	Sokal and Sneath(1984)			
7	4b				
8	4c				

5 Experimental Results

In this section, the experimental results are analyzed and discussed. The TTD results of the experiments are studied by the full factorial ANOVA. The ANOVA result is shown in Table 2. From Table 2, we can see that the three main effects exhibit a statistical significance of less than 0.05. We also found that one of the three two-way interactions are significant at an α of 0.05. Table 3 gives us the TTD mean and the 95% confidence interval of each zone picking type. From Table 3, we can see that 4b zone picking type has the best performance.

The ANOVA in Table 2 indicates that the factor of clustering algorithm is significant at an α of 0.05. Table 4 gives us the TTD mean and the 95% confidence interval of each similarity coefficient. From Table 4, we can see that *Ochiai*'s similarity coefficient has the best performance, while Sokal's similarity coefficient has the worst performance.

Table 2. Full factorial ANOVA on the TTD results

Source	Sum of Squares	*df*	Mean Square	F	*p*
Main Effects					
Zone Type	5.01E+11	7	71557897185	1905.88112	0.000**
Similarity	1.32E+09	5	263851384.5	7.027447592	0.000**
Seed	8.99E+08	1	899052219.7	23.94545841	0.000**
Two-way Interactions					
Zone Type * Similarity	1.59E+09	35	45395619.24	1.209072053	0.186
Zone Type * Seed	5.13E+08	7	73346168.53	1.953510141	0.058*
Similarity * Seed	2.96E+08	5	59298640.59	1.579366694	0.162
Three-way Interactions					
Zone Type * Similarity * Seed	2.39E+09	35	68271490.24	1.818350586	0.002**

Note: One asterisk indicates significance at the 25 percent level, two 15 percent level, three 10 percent level, and four the 5 percent level or below. This notation will be used throughout this paper.

Table 3. The TTD mean and the 95% confidence interval of each Zone Picking Type

Similarity Coefficient	Mean	95% Confidence Interval	
		Lower Bound	Upper Bound
2a	72012.725	71379.488	72645.962
2b	74842.631	74209.393	75475.868
3a	52469.108	51835.871	53102.346
3b	49108.003	48474.765	49741.240
3c	51517.775	50884.538	52151.012
4a	41717.608	41084.371	42350.846
4b	38579.144	37945.907	39212.382
4c	38749.792	38116.554	39383.029

The ANOVA in Table 2 indicates that the two-way interaction between the zone picking types and the batching method is significant at an α of 0.05. Table 5 summarize the TTD means of SNVL and SNVA. To understand the effects that batching method have on the TTD performance of various zone picking types, we conducted a

Duncan test on the TTD performance of zone picking types under SNVL and SNVA (see Table 6). From Table 6, the performance rankings of zone picking types under SNVL and SNVA are different, and their subset groupings of zone picking types are not identical either.

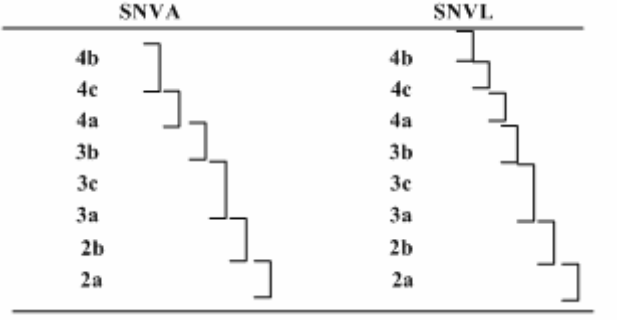
Table 4. The TTD mean and the 95% confidence interval of each similarity coefficient

Similarity Coefficient	Mean	95% Confidence Interval	
		Lower Bound	Upper Bound
SC1	51760.773	51212.373	52309.173
SC2	51605.488	51057.088	52153.887
SC3	51744.169	51195.769	52292.568
SC4	53189.596	52641.196	53737.996
SC5	52988.506	52440.107	53536.906
SC6	52959.058	52410.659	53507.458

Table 5. The TTD mean and the 95% confidence interval of each order batching method

Route-Planning Method	Mean	95% Confidence Interval	
		Lower Bound	Upper Bound
SNVL	8 8	2 0	2 2
SNVA	2 208	2 02	2

Table 6. The Duncan-test result on the TTD performance of Zone Picking type under SNVA and SNVL



Note: Zone Picking type connected symbolically are not significant at an α of 0.05.

6 Summary and Conclusions

In this paper, we studied the problem of converting a regular warehouse into various geometric zone-picking warehouses. We proposed a two-stage solution procedure to solve this problem. At the first stage, we adopted six similarity coefficients to meas-

ure the order-similarity between items. At the second stage, clustering algorithms are applied to form item groups. The number of item groups is equal to the various number of zones (e.g., 2,3,4) that one would like to have in the warehouse. The cumulative clustering algorithms are proposed. After that, each zone is assigned with an item group. Items belonging to the same item group will be stored in the picking zone respectively. The second problem is to determine the storage locations of all items in each zone. For this problem, we proposed two storage-location assignment rules. Since it is assumed that order batching is adopted in the warehouse, we adopted order-batching methods in the experiments. These two order-batching methods are different in their seed-order selection rules. Finally, the LG+SA method is used to optimize the routing-policy. Totally there are 96 (6•8•1•2•1) combinations of similarity coefficients, various geometric zone picking types, clustering algorithms, seed-order selection rules and route-planning methods. Each combination is tested with thirty randomly generated test problems. Their solution results are collected and analyzed to understand each method's performance and the effects that they have on each other's TTD performance. We also identified the best and the worst combinations of similarity coefficient, zone picking types, and seed-order selection rule in terms of their TTD performance. From the results, we can conclude that a significant amount of improvement can be achieved in minimizing TTD if appropriate zone-division is applied to the logistic warehouse.

References

1. Armstrong, R.D., Cook, W.D., Saipe, A.L.: Optimal batching in a semi-automated order-picking system. Journal of the Operational Research Society 30(8), 711–720 (1979)
2. Coyle, J.J., Bardi, E.J., Langley, C.J.: The Management of Business Logistics. West, St. Paul MN (1996)
3. De Koster, R., Van der Poort, E.: Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions. IIE Transactions 30, 469–480 (1998)
4. De Koster, M.B.M., Van der Poort, E.S., Wolters, M.: Efficient orderbatching methods in warehouses. International Journal of Production Research 37(7), 1479–1504 (1999)
5. Elsayed, E.A.: Algorithms for optimal material handling in automatic warehousing system. International Journal of Production Research 19(5), 525–535 (1981)
6. Elsayed, E.A., Stern, R.G.: Computerized algorithms for order processing in automated warehousing systems. International Journal of Production Research 21(4), 579–586 (1983)
7. Gibson, D.R., Sharp, G.P.: Order batching procedures. European Journal of Operational Research 58(1), 57–67 (1992)
8. Gray, A.E., Karmarkar, U.S., Seldmann, A.: Design and operation of an order-consolidation warehouse models and application. European Journal of Operational Research 59, 3–13 (1992)
9. Hall, R.W.: Distance approximations for routing manual pickers in a warehouse. IIE Transactions 25(4), 76–87 (1993)
10. Hausman, W.H., Schwarz, L.B., Graves, S.C.: Optimal storage assignment in automatic warehousing systems. Management Science 22(6), 629–638 (1976)
11. Ho, Y.C., Chien, S.P.: A comparison of two zone-visitation sequencing strategies in a distribution centre. Computers and Industrial Engineering 50, 426–439 (2006)

12. Ho, Y.C., Tseng, Y.Y.: A study on the order-batching problem of order-picking in distribution centers. In: Proceedings of The Fifth Asia-Pacific Industrial Engineering and Management Systems Conference & The Seventh Asia-Pacific Division Meeting of the International Foundation of Production Research, December 12-15, ANA Hotel, Gold Coast, Australia (2004)
13. Hwang, H., Baek, W., Lee, M.K.: Clustering algorithms for order picking in an automated storage and retrieval system. International Journal of Production Research 26(2), 189–201 (1988)
14. Hwang, H., Lee, M.K.: Order batching algorithms for a man-on-board automated storage and retrieval system. Engineering Costs and Production Economics 13(4), 285–294 (1988)
15. Jarvis, J.M., McDowell, E.D.: Optimal product layout in an order picking warehouse. IIE Transactions 23(1), 93–102 (1991)
16. Petersen, C.G: An evaluation of order picking routeing policies. International Journal of Operations and Production Management 17(1), 1098–1111 (1997)
17. Petersen II, C.G., Schmenner, R.W.: An evaluation of routing and volume-based storage policies in an order picking operation. Decision Sciences 30(2), 481–501 (1999)
18. Ratliff, H.D., Rosenthal, A.S.: Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem. Operations Research 31(3), 507–521 (1983)
19. Romesburg, H.C.: Cluster Analysis for Researchers, Lifetime Learning Publications (Wadsworth Inc.), Belmont, CA (1984)
20. Roodbergen, K.J., De Koster, R.: Routing methods for warehouses with multiple cross aisles. International Journal of Production Research 39(9), 1865–1883 (2001)
21. Rosenwein, M.B.: A comparison of heuristics for the problem of batching orders for warehouse selection. International Journal of Production Research 34(6), 657–664 (1996)
22. Sarker, B.R.: The resemblance coefficients in group technology: a survey and comparative study of relational matrices. Computers and Industrial Engineering 30(1), 103–116 (1996)
23. Schwarz, L.B., Graves, S.C., Hausman, W.H.: Scheduling policies for automatic warehousing systems: simulation results. AIIE Transactions 10(3), 260–270 (1978)
24. Thompkins, J.A., White, J.A., Bozer, Y.A., Frazelle, E.H., Tanchoco, J.M.A., Trevino, J.: Facilities planning. John Wiley, New York (1996)

Routing Path Generation for Reliable Transmission in Sensor Networks Using GA with Fuzzy Logic Based Fitness Function*

Jin Myoung Kim and Tae Ho Cho

School of Information and Communication Engineering, Sungkyunkwan University
300 Cheoncheon-dong, Jangan-gu, Suwon 440-746, Korea
{kjm77,taecho}@ece.skku.ac.kr

Abstract. There has been a growing interest in the applications of sensor networks, such as combat field surveillance, terrorist tracking and highway traffic monitoring. These applications gather sensed data from sensor nodes to monitor events in the territory of interest. One of the important issues in these applications is the existence of the radio-jamming zone between source nodes and the base station. Depending on the routing protocol the transmission of the sensed data may not be delivered to the base station. To solve this problem we propose a routing path generation method that is based on genetic algorithms for reliable transmission by considering the balanced energy depletion of the sensor nodes. The genetic algorithm finds an efficient routing path by considering the radio-jamming zone, energy consumption needed for data transmission and average remaining energy. The fitness function employed in the genetic algorithm is implemented by applying the fuzzy logic. In simulation, our proposed method is compared with LEACH and Hierarchical PEGASIS. The simulation results show that the proposed method is efficient in both the energy consumption and success ratio of the delivery.

Keywords: Genetic algorithm, shortest path, sensor network, radio-jamming zone, fuzzy logic.

1 Introduction

Recent advances in MEMS (micro-electro-mechanical systems) and low power highly integrated digital electronics have enabled the development of low-cost sensor networks [1][2]. That is, WSNs are composed of small autonomous devices or sensor nodes that are networked together [3][4].

Many of the events and problems that occur in deployed WSNs exhibit a spatio temporal property. For example, an attacker deploys malicious nodes in a network, which conduct radio jamming in order to interrupt communication of the sensor nodes

* This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement). (IITA-2006-C1090-0603-0028).

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 637–648, 2007.
© Springer-Verlag Berlin Heidelberg 2007

[5]. Applications of WSNs collect sensed data from sensor nodes to monitor events in the interest territory. Therefore, it is important that sensed data in the territory is sent to the destination.

In general, sensor nodes are disposable and expected to last until their energy drains. Therefore, energy consumption needs to be managed in order to allow sensor nodes to operate for a specific minimum duration [6].

In this paper, we propose a routing path generation method that is based on genetic algorithm (GA) for the reliable data transmission by considering the radio-jamming zone, energy consumption needed for the data transmission and average remaining energy. There are many algorithms for finding a routing path, such as the Dijkstra algorithm and the Bellman-Ford algorithm [7]. These algorithms can find a routing path in polynomial time. But they are unsuitable for WSNs [8]. The GA promises to be a solution to real-time communication of rapidly changing topology in sensor networks. Also, the hardware implementation of the GA is extremely fast [9]. The fitness function employed in the GA is implemented by applying the fuzzy logic. The fuzzy logic is exploited to estimate a threshold. In simulation, the proposed method is compared with existing method LEACH [10] and Hierarchical PEGASIS [11]. The simulation results show that the proposed method guarantees sensed data transmission from cluster to base station.

The remainder of the paper is organized as follows. Section 2 provides an overview and discusses the motivation of this research. Section 3 shows the details of the routing method using GA. Section 4 represents the design of GA. Section 5 shows fuzzy rules that are employed in the fitness function of GA. Section 6 reviews the simulation results and compares the proposed method with an existing method. Conclusions are presented in section 7.

2 Motivation

There are many types of routing protocols in wireless sensor networks. Sensed data in the network is sent to the base station using a routing method. An attacker interrupts data transmission. In this section, we discuss the routing method and the motivation.

2.1 Overview

WSNs are composed of hundreds or thousands of sensor nodes. Individual sensor nodes have limited computation, are immobile and cannot be recharged.

In general, routing in WSNs can be classified into three types. These are flat-based routing, clustering-based routing, and direct communication-based routing. In flat-based routing, all of the nodes have the same role, whereas in clustering-based routing, the transmission of messages from nodes is done through the cluster head (CH), resulting in two different types of nodes. In direct communication-based routing, a sensor node sends data directly to the base station (BS) [12][13]. In this protocol, if the diameter of the network is large, the power of the sensor nodes will be drained very quickly. In flat protocols, when a node needs to send data, it may find a route consisting of several hops to the BS [12]. Clustering-based routing is an efficient way

to lower energy consumption within a cluster, performing data aggregation and fusion in order to decrease the number of messages transmitted to the BS [13].

2.2 Motivation

In a hierarchical routing protocol, a CH organizes a cluster. If the BS sends the data request message to the CH, the CH collects the data from sensor nodes within the cluster and sends the data to the BS.

Figure 1 shows data transmission from the CH to the BS. In figure 1, the CH_1 directly sends the data to the BS. But the CH_2, which is interrupted by an obstruction, cannot directly send the data to the BS. There are various types of obstructions, such as geographical quality, radio-jamming zone by attacker and so on. The obstruction in this paper is the radio-jamming zone. The attacker creates the radio-jamming zone or wormhole by deploying malicious nodes specifically to interrupt the data transmission [14]. The detailed explanation of the radio-jamming zone can be found in [5].

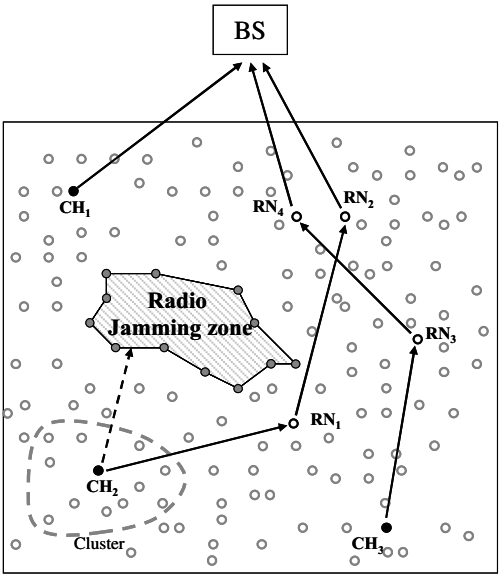


Fig. 1. Routing method in sensor network to ensure data transmission and balanced energy depletion

The CH_2 cannot send the data to the BS through this jamming zone. Therefore, the sensed data in the CH_2 is transmitted to the BS by relaying nodes RN_1 and RN_2. The RN (Relay node) is a relayed sensor node.

Although the CH_3 is not interrupted by the radio-jamming zone, it can send data directly to the BS. If the CH_3 can not directly send data due to insufficient remaining energy, it sends data to the BS by relaying through nodes RN_3 and RN_4, which have sufficient remaining energy.

For reliable data transmission, we proposed a routing path generation method that is based on GAs. The GA module in the method generates a routing path by considering the radio-jamming zone, energy consumption needed for data transmission and average remaining energy. We assume that the BS is aware of the remaining energy level and the location of sensor nodes. It is also assumed that the sensor nodes are able to control the distance of transmission range.

3 GA Based Routing Method

In our proposal, the remaining energy level of the sensor nodes and radio-jamming zone are subject to frequent change. Accordingly, the routing path is decided by considering the location information of the radio-jamming zone and average remaining energy of sensor nodes.

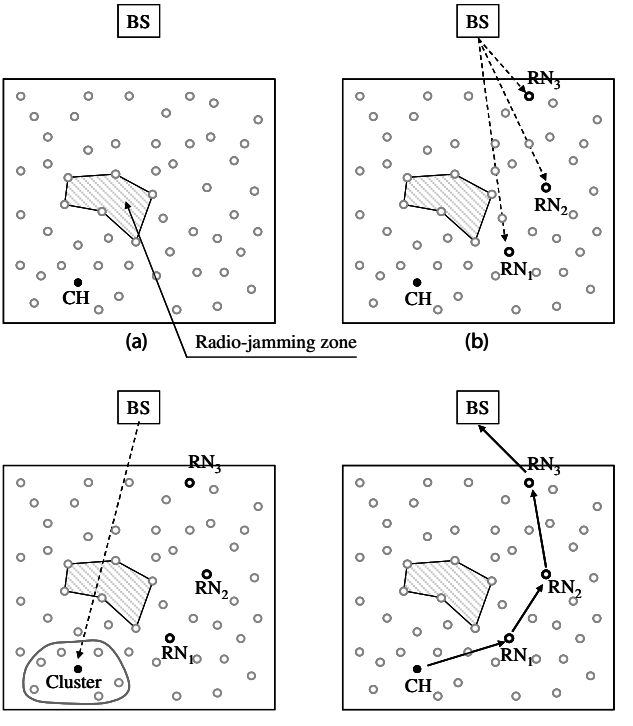


Fig. 2. Overview of GA based data transmission in sensor networks

Whenever the BS requires the sensed data in the territory, the routing path is generated using the GA based routing method.

The figure 2 shows the step of the data transmission. Figure 2 (a) shows the decision of the cluster head needed to collect the data. Figure 2 (b) shows that the BS finds a routing path using GA, after the decision of the cluster head. The routing path

is CH→RN_1→RN_2→RN_3→BS. The BS sends a message to the nodes RN_1, RN_2 and RN_3 needed for the data transmission. This message includes the MAC (Message Authentication Code) of the next node, and the radio strength needed for data transmission to the next node, and is saved in the memory of the sensor node. This information is removed from the memory of the sensor node when the data is sent to the next node.

For example, when the BS sends a message to the node RN_1, this message includes the MAC of the node RN_2 and the radio strength needed for data transmission to the node RN_2. Accordingly, the nodes RN_1, RN_2 and RN_3 receive the information needed for data transmission to the next node or BS.

Figure 2 (c) shows that the BS sends a request message to the cluster head, after finding the routing path. This message includes the MAC of the node RN_1 and radio strength needed for data transmission to the node RN_1. Figure 2 (d) shows that the collected data in the cluster head can be transmitted to the base station by relaying through nodes RN_1, RN_2 and RN_3.

4 Design of Genetic Algorithm

In general, the GA provides an efficient manner to search for an optimal solution. The algorithm starts by randomly generating an initial population of possible solutions.

4.1 Genetic Presentation

A chromosome represents a routing path. Each gene in the chromosome is represented by a positive integer. The gene corresponds to a sensor node. These positive integers represent the IDs of sensor nodes. Each locus of the chromosome represents the order of a node in a routing path.

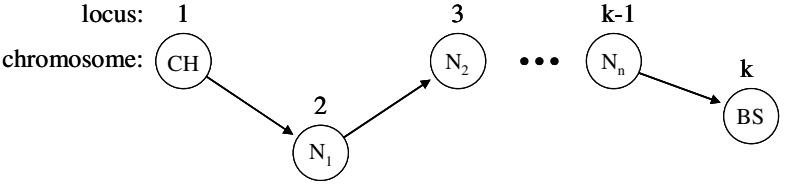


Fig. 3. Routing path and encoding scheme

The gene of the first and last locus is always reserved for the CH and BS, respectively. The genes between the first and last locus represent relay nodes. The length of the chromosome is variable, but it should not exceed the number of sensor nodes in the sensor network. Figure 3 is an example of chromosome encoding from the cluster head to the base station.

4.2 Crossover

The crossover examines the current solutions in order to find better ones. Figure 4 shows the crossover procedure [15].

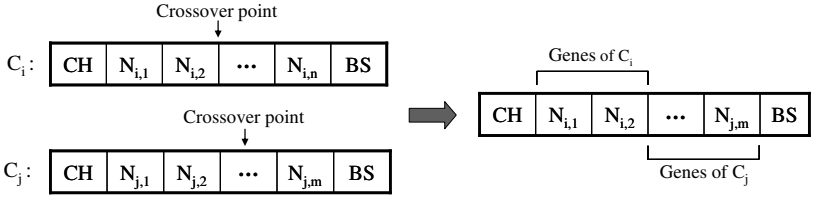


Fig. 4. Procedure of the crossover

In figure 4, the crossover point of the C_i and C_j is decided by a random number between 0 and n, and between 0 and m, respectively. Two partial chromosomes C_i and C_j are assembled in order to generate a new chromosome.

4.3 Selection

The selection operation is used to improve the average quality of the population. This operation improves the chance that the high quality chromosome is selected as the next generation. In this paper, we use a tournament mechanism. Two chromosomes are elected and the best one is selected.

4.4 Mutation

A mutation is needed for keeping away from local optima. The mutation raises an infinitesimal increase in the probability of inducing bias. In this paper, the mutation occurs in the selection operation. When two chromosomes are picked in the selection operation, one that is not fitter is selected with 1 percent mutation probability.

4.5 Fitness Function Using Fuzzy Logic

The fitness function must accurately measure the quality of the chromosomes in the population. As shown in section 4.1, the chromosome represents a routing path. To evaluate the fitness of the chromosome, the fitness function is implemented by applying the fuzzy logic (Figure 5).

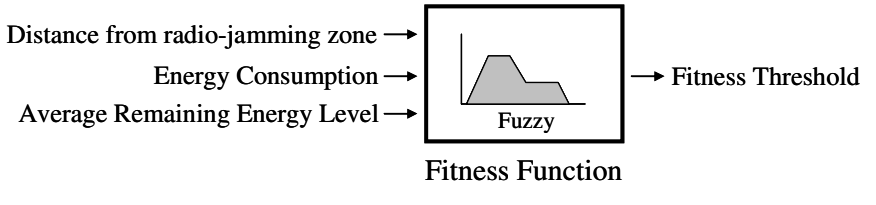


Fig. 5. The fitness function employed in the GA is implemented by applying the fuzzy logic

The distance from the routing path to the radio-jamming zone, energy consumption needed for the data transmission, and average remaining energy are used to evaluate the fitness of the chromosome. The membership function is defined as follows:

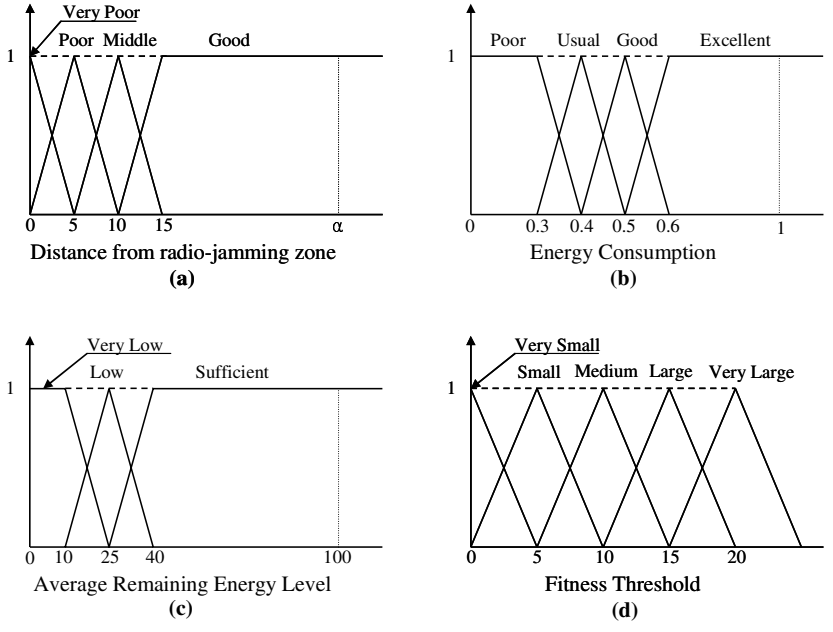


Fig. 6. Membership functions – the distance from radio-jamming zone (a), energy consumption (b), average remaining energy level (c) and fitness threshold (d)

In the figure 6 (a), the distance from the radio-jamming zone is the shortest distance between the routing path and radio-jamming zone. The α that is decided by a network operator is the length of the territory of interest. In the figure 6 (b), the energy consumption that is normalized is the energy needed for the data transmission. The figure 6 (c), the average remaining energy level is the average remaining energy of the nodes in the chromosome. The fitness threshold in the figure 6 (d) is the estimate of the fitness of the chromosome.

In the next section, we describe fuzzy rules for some occasions.

5 Fuzzy Rules in Fitness Function

If the genes in the chromosome are nearly located in the radio-jamming zone, the fitness threshold is very small (e.g., 0).

```
R04: if Distance from radio-jamming zone is Very Poor
     and Energy Consumption is Excellent
     and Average Remaining Energy Level is Sufficient
     then Fitness Threshold is Very Small
```

If the energy consumption needed for the data transmission from the CH to the BS by the chromosome is more three times than the energy consumption need for the data transmission from the CH to the BS by single hop, the fitness threshold is very small.

```
R07: if Distance from radio-jamming zone is Good
     and Energy Consumption is Poor
     and Average Remaining Energy Level is Sufficient
     then Fitness Threshold is Very Small
```

If the average remaining energy is less then 25% and energy consumption needed for the data transmission from the CH to the BS by the chromosome is similar with the energy consumption needed for the data transmission from the CH to the BS by the single hop, the fitness threshold is small.

```
R15: if Distance from radio-jamming zone is Good
     and Energy Consumption is Good
     and Average Remaining Energy Level is Very Low
     then Fitness Threshold is Small

R27: if Distance from radio-jamming zone is Good
     and Energy Consumption is Usual
     and Average Remaining Energy Level is Very Low
     then Fitness Threshold is Very Small
```

6 Simulation Results

To shows the effectiveness of proposed method, we have compared the proposed method with LEACH and Hierarchical PEGASIS (H-PEGASIS). In this simulation, 100 sensor nodes are randomly placed in a 100m x 100m territory. The base station is located in (50m, 150m). The shape of the radio-jamming zone is a circle. Each node is equipped with an energy source, where the total amount of energy accounts for 0.25J at the beginning of the simulation. Each node takes 16.25, 12.5μJ to transmit/receive one byte per meter. The packet size is 32bytes.

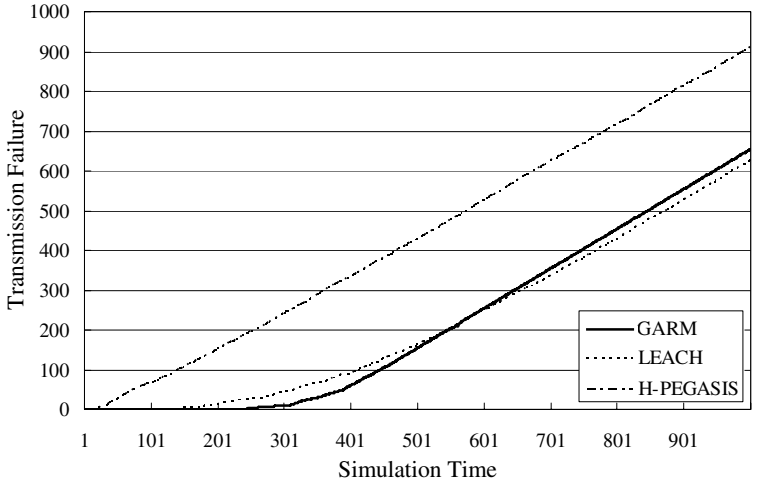


Fig. 7. The increase of the data transmission failures caused by energy insufficiency of sensor nodes

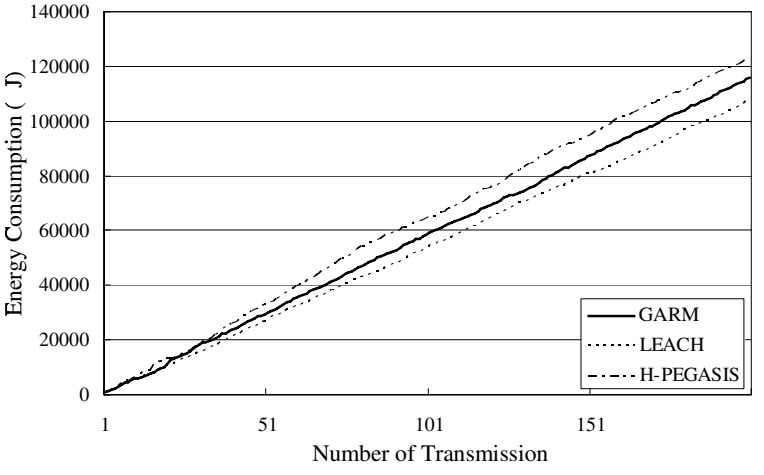


Fig. 8. The increase of the energy consumption with change of the number of the data transmission

Figure 7 illustrates the simulation results of the increase of the data transmission failure caused by the energy insufficiency. Since the proposed method finds a routing path by considering the average remaining energy, the number of the data transmission failures in the proposed method is less than H-PEGASIS in all range. Also the proposed method consumes less energy than LEACH between 1 and 580. In upper 580, however, the proposed method consumes more energy than LEACH. That is explained in figure 8.

Figure 8 shows the increase of consumption caused by the data transmission. In the figure, our proposed method consumes less energy than H-PEGASIS. Since the one of the considered factors in the proposed method is the energy consumption, the energy consumption by a data reception in the proposed method is less than H-PEGASIS. The proposed method consumes more energy than LEACH, since the data in the proposed method can be sent to the base station in both single or multi hop fashion for reliable data transmission.

In figure 9 and 10, we measured the ratio of the data transmission failure caused by the radio-jamming zone when the data transmission occurs 1000 times. Figure 9 and 10 show the ratio of the data transmission failure with change the size and location of the zone respectively.

In figure 9, the location of radio-jamming zone is (70m, 30m). In the figure, the ratio of the data transmission failure in the proposed method is 0%, since the proposed method finds a routing path by considering the radio jamming for reliable data transmission. The small, medium and large in the figure denote the zone size with the radius 5m, 10m and 15m respectively. The proposed method is also not affected by the zone size. Figure 9 shows that the radio of the data transmission in LEACH and H-PEGASIS increase with the zone size.

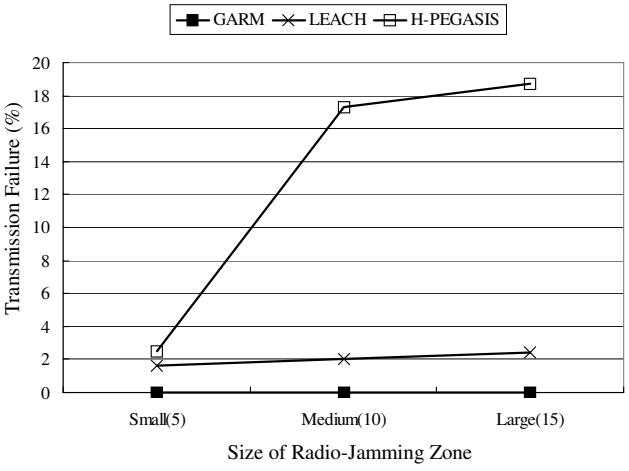


Fig. 9. The ratio of the data transmission failure caused by the radio-jamming zone that is located in (70m, 30m)

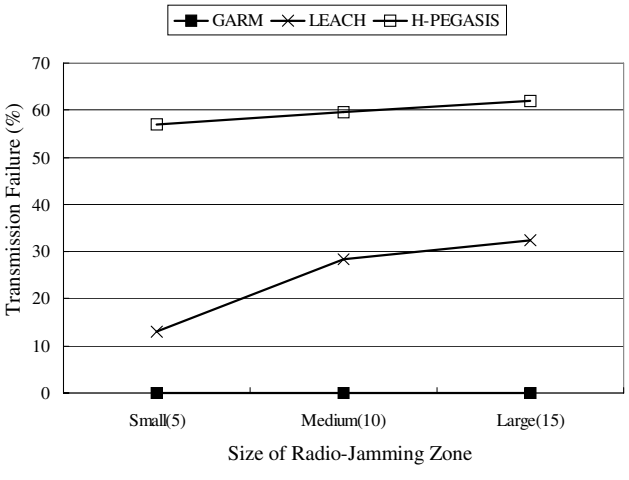


Fig. 10. The radio of the data transmission failure caused by the radio-jamming zone that is located in (35m, 70m)

Figure 10 shows the ratio of the data transmission failure caused by the radio-jamming zone that is located in (35m, 70m). The location of the BS is (50m, 150). So, the simulation result shown in figure 10 has the radio-jamming zone closer to the BS.In the figure, the proposed method is not affected by the location of the zone. But LEACH and H-PEGASIS dramatically increase with the location of zone.

7 Conclusion

The main purpose of WSNs is to collect the sensed data in the target system or application. It is important to guarantee that sensed data is sent to the destination. In this paper, we propose a routing method that guarantees data transmission from the cluster head to the base station, and operates balanced energy consumption. In the proposed method, the routing path is determined by considering the energy consumption needed for the data transmission, average remaining energy and radio-jamming zone. We apply the genetic algorithm to find the routing path. Also, in the genetic algorithm, we implement a fitness function using fuzzy logic.

Simulation results show that the data transmission from the cluster head to the base station is guaranteed in our proposed method. The result also shows that the proposed method can collect much more data in the territory than the existing method. Our method applies to applications that need reliable data transmission, such as battlefield and terrorist tracking.

In future research, we will improve the crossover operation using a heuristic algorithm by considering the network topology. We will also research into the optimal population size.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A Survey on Sensor Networks. IEEE Commun. Mag. 40(8), 102–114 (2002)
2. Akkaya, K., Younis, M.: A Survey on Routing Protocols for Wireless Sensor Networks. Ad hoc Netw. 3(3), 325–349 (2004)
3. Eltoweissy, M., Wadaa, A., Olariu, S., Wilson, L.: Group Key management scheme for large-scale sensor networks. Ad Hoc Network 3, 668–688 (2005)
4. Eschenauer, L., Virgil Gligor, D.: A Key Management Scheme for Distributed Sensor Networks. In: ACM CCS'02, pp. 41–47 (2002)
5. Wood, A.D., Stankovic, J.A., Son, S.H.: JAM: A Jammed-Area Mapping Service for Sensor Networks. IEEE Real-Time Systems, pp. 286–297 (2003)
6. Younis, M., Youssef, M., Arisha, K.: Energy-Aware Routing in Cluster-based Sensor Networks. In: IEEE MASCOTS, pp. 129–136 (2002)
7. Stalling, W.: High-Speed Networks: TCP/IP and ATM Design Principles. Prentice-Hall, Englewood Cliffs (1998)
8. Ahn, C.W., Ramakrishna, R.S.: A Genetic Algorithm for Shortest Path Routing Problem and Sizing of Populations. IEEE Transaction on Evolutionary Computation 6, 566–579 (2002)
9. Tufte, G., Haddow, P.C.: Prototyping a GA pipeline for complete hardware evolution. In: 1st NASA/DoD Workshop on Evolvable Hardware, pp. 76–84 (1999)
10. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on Wireless Communications 1(4), 660–670 (2002)
11. Lindsey, S., Raghavendra, C.S., Sivalingam, K.: Data Gathering in Sensor Networks using the Energy*Delay Metric. IPDPS 2001, pp. 2001–2008 (2001)

12. Jiang, Q., Manivannan, D.: Routing protocols for sensor networks. CCNC, pp. 93–98 (2004)
13. Al-Karaki, J.N., Kamal, A.E.: Routing techniques in wireless sensor networks: a survey. Wireless Communications 11, 6–28 (2004)
14. Li, Z., Trappe, W., Zhang, Y., Nath, B.: Robust Statistical Methods for Securing Wireless Localization in Sensor Networks. In: IPSN, pp. 91–98 (2005)
15. Hue, X.: Genetic algorithms for optimization: Background and application. Edinburgh Parallel Computing Centre 10 (1997)

A Heuristic Local Search Algorithm for Unsatisfiable Cores Extraction*

Jianmin Zhang, Shengyu Shen, and Sikun Li

School of Computer Science, National University of Defense Technology
410073 Changsha, China
{jmzhang,syshen,skli}@nudt.edu.cn

Abstract. Explaining the causes of infeasibility of Boolean formulas has many practical applications in various fields. We are generally interested in a small explanation of unsatisfiability that excludes irrelevant information. A small unsatisfiable core provides a succinct explanation of infeasibility and is valuable for applications. In recent years the problem of finding unsatisfiable subformulas has been addressed frequently by research works, which are mostly based on the SAT solvers with DPLL backtrack-search algorithm. However little attention has been concentrated on extraction of unsatisfiable cores using incomplete methods. In this paper, we propose a heuristic-based local search algorithm to derive unsatisfiable cores. This approach directly constructs the resolution sequence for proving unsatisfiability with a local search procedure, and then extracts small unsatisfiable cores from the sequence. We report and analyze the experimental results on benchmarks.

Keywords: Local search, Unsatisfiable core, Unit clause propagation, Binary clause resolution, Equality reduction.

1 Introduction

Many real-world problems, arising in formal verification, artificial intelligence and electronic design, can be formulated as constraint satisfaction problems, which are translated into Boolean formulas in conjunctive normal form (CNF). Boolean satisfiability (SAT) solvers, such as Chaff [1] and MiniSAT [2], are usually able to determine whether a large formula is satisfiable or not. When a formula is unsatisfiable, it is often required to find an unsatisfiable core, that is, a small unsatisfiable subset of the original formula. Localizing an unsatisfiable core is necessary to determine the underlying reasons for the failure. Explaining the causes of infeasibility of Boolean formulas is an essential requirement in many fields. A paradigmatic example is SAT-based model checking on predicate abstraction, where analysis of unsatisfiability is a necessary step for ensuring completeness of bounded model checking [3]. Additional examples include repairing inconsistent knowledge from a knowledge base [4], counterexample minimization

* This work is supported by the National Natural Science Foundation of China under grant No. 60603088.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 649–659, 2007.
© Springer-Verlag Berlin Heidelberg 2007

and explanation [5], and fixing wire routing in FPGAs [6]. Furthermore, we are usually interested in a small explanation of infeasibility that excludes irrelevant information. There have been considerable researches in deriving the unsatisfiable cores. Most of previous works are complete search approaches, essentially on the basis of enhanced versions of the Davis-Putnam-Logemann-Loveland (DPLL) backtrack-search algorithm. However, existing studies have very little concern regarding unsatisfiable cores extraction using incomplete local search method.

In this paper, we tackle the problem of finding small unsatisfiable cores from Boolean formulas by a heuristic local search algorithm. This approach is the first work we are aware of to adopt resolution-based stochastic local search method to extract unsatisfiable cores. Firstly, a local search procedure is employed to compute the resolution sequence for proving unsatisfiability of a formula. The process of deriving an empty clause is combined with some reasoning heuristics, such as unit clause propagation, binary clause resolution and equality reduction. Then each resolution sequence is constructed as a tree, and the unsatisfiable core is composed of all the leaves of this tree, because the original clauses involved in the derivation of the empty clause are referred to as the unsatisfiable core. Finally, a recursive function is used to find all of the leaves which correspond to the original clauses, and then a small unsatisfiable core is obtained.

The paper is organized as follows. The next section surveys the related work on computing unsatisfiable cores. Section 3 introduces the basic definitions and notations used throughout the paper. Section 4 proposes the local search algorithm for extracting small unsatisfiable cores from a Boolean formula in CNF. Section 5 presents some heuristics to improve the efficiency of our algorithm. Section 6 shows and analyzes experimental results on well-known pigeon hole problem instances. Finally, Section 7 concludes the paper and outlines future research work.

2 Related Work

There have been many different contributions to research on unsatisfiable cores extraction in the last few years, owing to the increasing importance in numerous practical applications. Some algorithms for deriving small unsatisfiable cores [7] are based on the ability of DPLL-based SAT solvers to produce resolution refutations. In [8], a method of adaptive core search guided by clauses hardness is employed to extract small unsatisfiable subformulas from the problem instances. In [9], an algorithm is suggested to compute a minimum unsatisfiable core. This approach essentially rely on enumerating all possible subsets of the formulas and checking which are unsatisfiable but all of whose subsets are satisfiable. Another algorithm with a similar search space, but intended to find all minimal unsatisfiable cores, is given in [10].

Another approach that allows for finding small unsatisfiable cores is called AMUSE [11], in which selector variables are added to each clause and the unsatisfiable core is derived by a branch-and-bound algorithm on the updated formula. A different algorithm that guarantees minimality is MUP [12], which is mainly a

prover of minimal unsatisfiability, as opposed to an unsatisfiable core extractor. Some research works, based on a strong relationship between maximal satisfiability and minimal unsatisfiability, have developed some sound techniques for finding all minimal unsatisfiable subformulas [13,14] or a minimum unsatisfiable core [15,16]. CoreTimmer [17] is also a method of deriving small unsatisfiable cores. It iterates over each internal node that consumes a large number of clauses and attempts to prove them without these clauses. In [18], a scalable algorithm, adopting a deeper exploration of resolution-refutation properties, is proposed for minimal unsatisfiable cores extraction from the formulas.

Experimental works on computing unsatisfiable cores can be grouped into complete search algorithms and incomplete search algorithms. Most of the above methods to extract unsatisfiable cores are complete search approaches, mainly on the basis of modern SAT solvers with DPLL backtrack-search procedure. In the recent past, a few research works have considered the problem of finding the unsatisfiable cores by incomplete local search algorithms.

In [19], the authors present an algorithm which tracks minimal unsatisfiable cores according to the trace of a failed local search run for consistency checking. However, this method is essentially based on a typical local search procedure for giving the Boolean formula a satisfiable interpretation. Recently, two distinct algorithms proposed in [20] are the first known works on using stochastic local search method for proving unsatisfiability. One of the approaches is to apply standard local search to a reformulation of the problem instance, and the other is a greedy randomized resolution algorithm that will eventually discover proofs of an unsatisfiable formula. Whereas up till now, no existing research work has concentrated on the unsatisfiable cores extraction from the proof of infeasibility utilizing a stochastic local search procedure.

3 Preliminaries

Resolution is a proof system for CNF formulas with the following inference rule:

$$\frac{(A \vee x)(B \vee \neg x)}{(A \vee B)} \ , \tag{1}$$

where A and B denote the disjunctions of literals. The clauses $(A \vee x)$ and $(B \vee \neg x)$ are the resolving clauses, and $(A \vee B)$ is the resolvent. The resolvent of the clauses (x) and $(\neg x)$ is the empty clause $(\perp)$. Every application of the inference rule is called a resolution step. A resolution sequence, namely a sequence of resolution steps, is that each one uses the result of the previous step or the clauses of the original formula as the resolving clauses of the current step.

Lemma 1. *A CNF formula φ is unsatisfiable iff there exists a finite sequence of resolution steps ending with the empty clause.*

It is well-known that a Boolean formula in CNF is unsatisfiable if it is possible to generate an empty clause by a finite sequence of resolution steps from the original clauses.

Definition 1. ***(Unsatisfiable Core).*** *Given a formula φ, ψ is an unsatisfiable core for φ iff ψ is an unsatisfiable formula and $\psi \subseteq \varphi$.*

Observe that an unsatisfiable core can be defined as any subset, which is infeasible, of the original formula. It is obvious that there may exist many different unsatisfiable cores with different numbers of clauses for the same problem instance, such that some of these cores are the subsets of others.

Lemma 2. *The set of original clauses involved in the derivation of the empty clause is referred to as the unsatisfiable core.*

That is to say, the clauses, contained in the intersection of a resolution sequence and the original formula, belong to some unsatisfiable core. Then we illustrate the process of extracting unsatisfiable cores from a Boolean formula according to Lemma 1 and Lemma 2. For example, the CNF formula is

$$\varphi = (x_1) \wedge (\neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3) \ . \tag{2}$$

The formula is refuted by a series of resolution steps ending with an empty clause. There are two resolution sequences to prove the infeasibility of this formula. One of the resolution sequences is

$$\frac{(x_1)(\neg x_1 \vee x_2)}{(x_2)} \longrightarrow \frac{(x_2)(\neg x_2)}{(\bot)} \ . \tag{3}$$

From the sequence, the resolvent (x_2) of the first resolution step serves as one of the resolving clauses of the second step, and the result of the second resolution step is an empty clause. According to Lemma 1, this formula is unsatisfiable. Therefore, the original clauses included in the proof of infeasibility belong to an unsatisfiable core. More specifically, the unsatisfiable core corresponding to the above resolution sequence is

$$\psi_1 = (x_1) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_2) \ . \tag{4}$$

Moreover, the other resolution sequence is

$$\frac{(x_1)(\neg x_1 \vee x_2)}{(x_2)} \longrightarrow \frac{(x_2)(\neg x_2 \vee x_3)}{(x_3)} \longrightarrow \frac{(x_3)(\neg x_3)}{(\bot)} \ . \tag{5}$$

This finite sequence of resolution steps arrives at an empty clause, and the unsatisfiable core consists of the original clauses involved in the process of resolution. Consequently, another unsatisfiable core is

$$\psi_2 = (x_1) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3) \ . \tag{6}$$

In conclusion, this simple example demonstrates that our stochastic local search algorithm to derive the small unsatisfiable cores is essentially based on Lemma 1 and Lemma 2.

4 Local Search for Unsatisfiable Cores Extraction

In recent years, the complete methods have made great progress in solving many real life problems including constraint satisfaction problem, but they usually cannot scale well owing to the extreme size of the search space. One way to solve the combinatorial explosion problem is to sacrifice completeness, thus some of the best known methods using this incomplete strategy are stochastic local search algorithms. In general, the local search strategy starts from an initial solution, which may be randomly or heuristically generated. Then the search moves to a better neighbor according to the objective function, and terminates if the goal is achieved or no better solution can be found. Stochastic local search methods are underlying some of the best-performing algorithms for certain types of problem instances, both from an empirical as well as from a theoretical point of view. Consequently, this strategy is adopted to tackle the problem of small unsatisfiable cores extraction from a Boolean formula. We propose a heuristic-based local search algorithm to compute unsatisfiable cores, based on Lemma 1 and Lemma 2. The algorithm, detailed in the later, is given as follows:

```
HLSAtoExtractSUC (formula)
   refuted = false
   iteration = 0
   while ((iteration < MAXITER) && !refuted) do
      if (Unit_Clause_Propagation() return UNSAT) then
         refuted = true
      else if (there exist binary clauses) then
         Binary_Clause_Resolution()
         Non_Tautology()
         Equality_Reduction()
         No_Same_Clause()
      else
         Randomly choose two clauses to resolve
      if (formula.size > MAXSIZE) then
         Remove a clause at random
      iteration++
   if (refuted == true) then
       print unsatisfiable
       SmallUC = Compute_Unsatisfiable_Core(sequence)
   else
       print unresolved
   return SmallUC
```

The local search algorithm begins with an input formula in CNF format. The objective function of this algorithm is to derive the empty clause, and a necessary condition for this to occur is that the formula contains at least some short clauses. We perform resolution of two clauses heuristically or randomly, until the formula is refuted or the upper limit of iterations is reached. In this

algorithm, the function called unit clause propagation can determine whether the formula is infeasible, because the formula is refuted if and only if an empty clause can be resolved by two unit clauses. The principle of unit clause propagation will be described in next section. If the current formula contains binary clauses, some reasoning strategies are employed in this algorithm, such as binary clause resolution and equality reduction, which will be given a detailed introduction later. The function named Non_Tautology fulfills the task of deleting the clauses which contain two opposite literals such as $(x_1 \vee \neg x_1 \vee x_3)$. The function of No_Same_Clause is to remove the duplicate clauses from the formula. These heuristics bring the powerful capacity of resolution on binary clauses. If there is no binary clause in the formula, two clauses will be randomly chosen to resolve according to the inference rule shown in Equation 1, and then the resolvent is added into the formula. However when the instance exceeds the maximum size constant, a clause is removed from the formula at random.

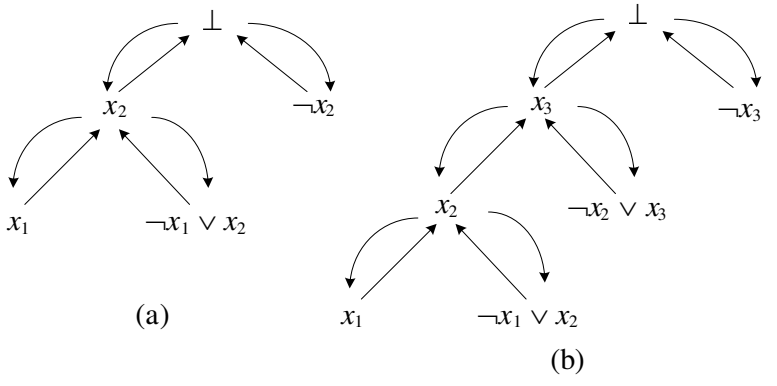


Fig. 1. The Process of Unsatisfiable Cores Extraction

While the algorithm is proceeding, we record the sequence of clauses engaged in the process of resolving an empty clause. Then a tree is constructed with respect to the resolution sequence. If the formula is refuted, a recursive function called Compute_Unsatisfiable_Core is employed to extract a small unsatisfiable core from the formation of a treelike arrangement. According to Lemma 2, we can conclude that all leaf nodes of the tree structure are actually referred to as the unsatisfiable core. Fig. 1 illustrates the process of finding small unsatisfiable cores from the formula denoted by Equation 2. As shown in Fig. 1, there are two trees respectively corresponding to two resolution sequences, which are represented as Equation 3 and Equation 5. The original clauses located on the leaves of a tree can be extracted by a recursive algorithm to form the unsatisfiable core. For example, in Fig. 1 (a), the root node, namely an empty clause, is resolved by an interim result (x_2) and a leaf node $(\neg x_2)$ which is an original clause. If we treat the clause (x_2) as a root node, the branches and leaves with the root also constitute a tree, and then the recursive function can be used on this subtree.

The clause (x_2) is resolved by two leaf nodes (x_1) and $(\neg x_1 \vee x_2)$. Therefore, an unsatisfiable core is composed of the three leaf clauses belonging to the original formula. Similarly, in Fig. 1 (b), another unsatisfiable core consists of the four leaf clauses $(\neg x_3)$, $(\neg x_2 \vee x_3)$, (x_1), and $(\neg x_1 \vee x_2)$.

5 Some Heuristics in the Algorithm

To improve the efficiency of the local search algorithm, we implement some reasoning heuristics in the algorithm. One of the heuristics is unit clause propagation, which is widely used to simplify a Boolean formula in DPLL-based SAT solvers. A so-called unit clause is the clause with the length 1, that is, it only contains one literal. Unit clause propagation selects a unit clause from the original formula, and then performs the reduction on the formula by the literal in the unit clause. We achieve this reduction in two kinds of situation: Firstly, if some clause contains a literal which is negative of the literal in the unit clause, the corresponding literal is deleted from that clause; Secondly, we eliminate the clauses which include the literal of the unit clause. We demonstrate the process of unit clause propagation by an example. Considered the CNF formula shown in Equation 2, the clause $(\neg x_2)$ is a unit clause, and is propagated to the whole formula. In accordance with the reduction rule of unit clause propagation, the literal (x_2) is removed from the third clause $(\neg x_1 \vee x_2)$, and the fourth clause $(\neg x_2 \vee x_3)$ is deleted. Consequently, the formula is turned into

$$\varphi' = (x_1) \wedge (\neg x_2) \wedge (\neg x_1) \wedge (\neg x_3) \ . \tag{7}$$

After applying the unit clause propagation, one can observe that the formula is strongly simplified and easily refuted. Furthermore, because unit clause propagation might generate new unit clauses, it is an iterative process of executing reductions by unit clauses until either of the following condition is reached: one is that an empty clause is resolved, or the other is that there are no more unit clauses in the remain formula. Unit clause propagation is the first phase of formula reduction used in the local search algorithm. The order in which the unit clause reductions occur is not important to the correctness of our local search algorithm. It can be proved that applying unit clause propagation on the formula do not change the final result of the resolution sequence.

In general, a Boolean formula might also have many binary clauses, which are defined as the clauses including two literals. Then it is possible to do a lot of reductions on the original formula by reasoning with these binary clauses as well. The reductions of two binary clauses arise if and only if they contain one pair of opposite literals, and abide by the inference rule depicted in Equation 1. For instance, a Boolean formula in CNF is given as follows:

$$\phi_1 = (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (x_1 \vee x_3) \ . \tag{8}$$

There are three binary clauses, which can be resolved by the inference rule, in this formula. Then the process of resolution between the binary clauses is

$$\frac{(\neg x_1 \vee x_2)(\neg x_2 \vee x_3)}{(\neg x_1 \vee x_3)}, \frac{(\neg x_1 \vee x_2)(x_1 \vee x_3)}{(x_2 \vee x_3)}, \frac{(\neg x_1 \vee x_3)(x_1 \vee x_3)}{(x_3)} \ . \tag{9}$$

Resolving the binary clauses produces the new binary clauses $(\neg x_1 \vee x_3)$, $(x_2 \vee x_3)$ and the new unit clauses (x_3). More generally, performing all possible resolutions of pairs of binary clauses can generate new binary clauses or new unit clauses. Further, this kind of reduction can tackle more complicated clauses than unit clause propagation. Therefore, binary clause resolution can be done in conjunction with unit clause propagation in a repeated procedure.

The third heuristic is equality reduction, which is also a type of useful binary clause reasoning mechanism. Equality reduction is essentially based on the following equation:

$$(x \Leftrightarrow y) = (x \Rightarrow y) \wedge (y \Rightarrow x) = (\neg x \vee y) \wedge (\neg y \vee x) \ . \tag{10}$$

If a formula contains two correlated clauses such as $(x \vee \neg y)$ and $(\neg x \vee y)$, we can form an updated formula by equality reduction. Equality reduction is a three-step procedure: Firstly, all instances of y in the formula are replaced by the literal x or vice versa; Secondly, all clauses containing both x and $\neg x$ are deleted; Finally, all duplicate instances of x or $\neg x$ are removed from all of the clauses. The process of equality reduction is illustrated by the following example. A Boolean formula in CNF is

$$\phi_2 = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_3) \ . \tag{11}$$

Obviously, one can conclude that x_1 is equivalent to x_2, because there exist $(x_1 \vee \neg x_2)$ and $(\neg x_1 \vee x_2)$ in the above formula. Then we substitute x_1 for x_2 throughout the formula, and perform reductions on the new clauses. Finally, we obtain the reduced formula:

$$\phi_2' = (x_1 \vee x_3) \wedge (\neg x_1 \vee x_3) \ . \tag{12}$$

Similar to binary clause resolution, such clause reasoning approach might yield new binary clauses. Consequently, equality reduction combined with unit clause propagation and binary clause resolution can run iteratively, until an empty clause is resolved or no new clause is added.

6 Experimental Results

To experimentally evaluate the effectiveness of our algorithm, we select 10 problem instances from the well-known pigeon hole family, and compare our algorithm with the greedy genetic algorithm [16] on this benchmark. The pigeon hole problem asks whether it is possible to place $n + 1$ pigeons in n holes without two pigeons being in the same hole. Obviously, it is impossible. Therefore, all of the problem instances are unsatisfiable. We choose these instances because they are well-known for having only one unsatisfiable core. Consequently, the

greedy genetic algorithm which extracts the minimum unsatisfiable core from a Boolean formula and our algorithm can obtain the same unsatisfiable core for one problem instance. The SAT encoding of the pigeon hole problem is very straightforward. For each pigeon i we have a variable x_{ij} which means that pigeon i is placed in hole j. With this way of encoding, the placing requirements are now formulated as a set of CNF clauses that fall into one of two categories: Liveness constraints are to ensure that each pigeon is assigned to at least one hole; Exclusivity constraints are to guarantee that each hole is assigned to at most one pigeon.

Our algorithm to extract small unsatisfiable cores is implemented in C++ using STL. The experiments were conducted on a 1.6 GHz Athlon machine having 1 GB memory and running the Linux operating system. The limit time was 3600 seconds. The experimental results are listed in Table 1. There are 10 instances, called "holex" corresponding to the problem of placing $x + 1$ pigeons in x holes, available at the benchmark set. Table 1 shows the number of clauses (clas) and the number of variables (vars) for each of the 10 problem instances. Table 1 also gives the number of clauses in the unique unsatisfiable core (UC size) for every instance. Furthermore, Table 1 provides the runtime in seconds of the greedy genetic algorithm (GGA time) to derive the unsatisfiable core. The last column presents the mean runtime of ten launches in seconds for the heuristic-based local search algorithm (HLSA time).

Table 1. Performance Results on Benchmark

Instances	**clas**	**vars**	**UC size**	**GGA time**	**HLSA time**
hole1	3	2	3	0	0
hole2	9	6	9	0	0
hole3	22	12	22	0	0
hole4	45	20	45	0	0
hole5	81	30	81	0.02	0
hole6	133	42	133	0.08	0.1
hole7	204	56	204	0.90	0.5
hole8	297	72	297	51.90	22.8
hole9	415	90	415	1304.00	682.6
hole10	561	110	561	time out	1850.0

From this table, we may observe the following. The heuristic-based local search algorithm outperforms the greedy genetic algorithm for most formulas, except for the instance of *hole6*. Our algorithm is able to successfully extract the unsatisfiable core at each launch. For the instance of *hole10*, the greedy genetic algorithm failed to find the unsatisfiable core within the timeout, but our algorithm succeeded in obtaining it. More generally, an incomplete method can solve the instances of a certain of problem class more effectively than complete

methods, such as 2-SAT and 3-SAT formulas. Our heuristic local search algorithm can efficiently solve these instances, partially because there are the great capabilities of reasoning short clauses, such as unit clause propagation, binary clause resolution and equality reduction, and the pigeon hole problem instances contain many binary clauses.

7 Conclusion

Explaining the causes of infeasibility of Boolean formulas has practical applications in many fields, such as formal verification, artificial intelligence and electronic design. A small unsatisfiable core provides a succinct explanation of infeasibility and is valuable for applications. In this paper, we present a heuristic-based local search algorithm to extract small unsatisfiable cores from CNF formulas. The algorithm is combined with some reasoning heuristics, including unit clause propagation, binary clause resolution and equality reduction. The experimental results illustrate that our algorithm outperforms the greedy genetic algorithm. The results also show that this local search algorithm can efficiently tackle the certain type of problem instances with many short clauses, and cannot work very well for the formulas with most long clauses, largely because it makes the decisions on resolution of two long clauses in a stochastic way, and lacks of the effective ways for selecting the right clauses. Therefore one of the future works is adding more aggressive heuristics for the resolution of long clauses. In other words, the future algorithm should select the appropriate resolving clauses, and find the short resolution sequence, because the shorter the sequence is, the smaller the unsatisfiable core is.

Acknowledgments. The authors would like to thank Steven Prestwich for sharing their implementation and discussion which helped to achieve our local search algorithm.

References

1. Moskewicz, M.W., Madigan, C.F., Zhao, Y., Zhang, L., Malik, S.: Chaff: engineering an efficient SAT solver. In: Proceedings of the 38th Design Automation Conference (DAC'01) (2001)
2. Een, N., Sorensson, N.: An extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, Springer, Heidelberg (2004)
3. McMillan, K.L., Amla, N.: Automatic abstraction without counterexamples. In: Garavel, H., Hatcliff, J. (eds.) ETAPS 2003 and TACAS 2003. LNCS, vol. 2619, Springer, Heidelberg (2003)
4. Mazure, B., Sais, L., Gregoire, E.: Boosting complete techniques thanks to local search methods. Annals of Mathematics and Artificial Intelligence 22(3-4), 319–331 (1998)
5. Shen, S., Qin, Y., Li, S.: Minimizing counterexample with unit core extraction and incremental SAT. In: Cousot, R. (ed.) VMCAI 2005. LNCS, vol. 3385, Springer, Heidelberg (2005)

6. Nam, G-J., Aloul, F., Sakallah, K., Rutenbar, R.: A comparative study of two Boolean formulations of FPGA detailed routing constraints. In: Proceedings of the 2001 International Symposium on Physical Design (ISPD'01) (2001)
7. Zhang, L., Malik, S.: Extracting small unsatisfiable cores from unsatisfiable Boolean formula. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, Springer, Heidelberg (2004)
8. Bruni, R., Sassano, A.: Restoring satisfiability or maintaining unsatisfiability by finding small unsatisfiable subformulae. In: Proceedings of the Fourth Workshop on Theory and Application of Satisfiability Testing (SAT'01) (2001)
9. Lynce, I., Marques-Silva, J.P.: On computing minimum unsatisfiable cores. In: Hoos, H.H., Mitchell, D.G. (eds.) SAT 2004. LNCS, vol. 3542, Springer, Heidelberg (2005)
10. de la Banda, M., Stuckey, P., Wazny, J.: Finding all minimal unsatisfiable subsets. In: Proceedings of the Fifth ACM-SIGPLAN International Symposium on Principles and Practice of Declarative Programming (PPDP'03) (2003)
11. Oh, Y., Mneimneh, M.N., Andraus, Z.S., Sakallah, K.A., Markov, I.L.: AMUSE: a minimally-unsatisfiable subformula extractor. In: Proceedings of the 41st Design Automation Conference (DAC'04) (2004)
12. Huang, J.: MUP: A minimal unsatisfiability prover. In: Proceedings of the Tenth Asia and South Pacific Design Automation Conference (ASP-DAC'05) (2005)
13. Liffiton, M.H., Sakallah, K.A.: On finding all minimally unsatisfiable subformulas. In: Bacchus, F., Walsh, T. (eds.) SAT 2005. LNCS, vol. 3569, Springer, Heidelberg (2005)
14. Bailey, J., Stuckey, P.J.: Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In: Proceedings of the Seventh International Symposium on Practical Aspects of Declarative Languages (PADL'05) (2005)
15. Mneimneh, M.N., Lynce, I., Andraus, Z.S., Marques-Silva, J.P., Sakallah, K.A.: A branch and bound algorithm for extracting smallest minimal unsatisfiable formulas. In: Bacchus, F., Walsh, T. (eds.) SAT 2005. LNCS, vol. 3569, Springer, Heidelberg (2005)
16. Zhang, J., Li, S., Shen, S.: Extracting minimum unsatisfiable cores with a greedy genetic algorithm. In: Proceedings of the 19th Australian Joint Conference on Artificial Intelligence (AUS-AI'06) (2006)
17. Gershman, R., Koifman, M., Strichman, O.: Deriving small unsatisfiable cores with dominator. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, Springer, Heidelberg (2006)
18. Dershowitz, N., Hanna, Z., Nadel, A.: A scalable algorithm for minimal unsatisfiable core extraction. In: Biere, A., Gomes, C.P. (eds.) SAT 2006. LNCS, vol. 4121, Springer, Heidelberg (2006)
19. Gregoire, E., Mazuer, B., Piette, C.: Tracking MUSes and strict inconsistent covers. In: Proceedings of the the Sixth Conference on Formal Methods in Computer-Aided Design (FMCAD'06) (2006)
20. Prestwich, S., Lynce, I.: Local search for unsatisfiability. In: Biere, A., Gomes, C.P. (eds.) SAT 2006. LNCS, vol. 4121, Springer, Heidelberg (2006)

ontoX - A Method for Ontology-Driven Information Extraction

Burcu Yildiz[1] and Silvia Miksch[1,2]

[1] Institute for Software Technology and Interactive Systems,
Vienna University of Technology, Vienna, Austria
{yildiz, miksch}@ifs.tuwien.ac.at
[2] Department of Information and Knowledge Engineering,
Danube University Krems, Krems, Austria
silvia.miksch@donau-uni.ac.at

Abstract. Information Extraction (IE) is an important research field within the Artificial Intelligence community, for it tries to extract relevant information out of vast amounts of data. In this paper, we propose an extraction method that utilises the content and pre-defined semantics of ontologies formulated in the Web Ontology Language (OWL) to perform the extraction task. We also propose our method to detect out-of-date constructs in the ontology to suggest changes to the user. After stating the results of our evaluation, we conclude that the use of ontologies in conjunction with IESs can indeed yield feasible results and contribute to the better scalability and portability of the system.

Keywords: Ontology-Driven Information Extraction, Information Extraction, Ontology Engineering.

1 Introduction

Since Berners-Lee proposed and started to endorse ontologies as the backbone of the Semantic Web in the nineties [1], a whole research field evolved around the fundamental engineering aspects of ontologies, such as the generation, evaluation, and management of ontologies. However, many researchers were curious about the use of ontologies within Information Systems (ISs) in 'ordinary' settings, performing 'ordinary' information processing tasks [2], where our focus lies on the use of ontologies within Information Extraction Systems (IESs) [3].

Information Extraction (IE) is defined as a form of natural language processing in which certain types of information must be recognised and extracted from text [4] [5]. It is difficult to describe what 'relevant information' actually is, and it is even harder to communicate this description to a computer system. For that purpose we analyse whether and how ontologies can be utilised to provide IESs with an unambiguous and formal description of relevant information. However, to enable the smooth integration of ontologies within IESs, several requirements have to be reconciled [6].

In this paper, we present a method for ontology-driven Information Extraction from natural language text, where the ontology is used as a formal and thus unambiguous "specification of the conceptualisation that represents the particular

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 660–673, 2007.
© Springer-Verlag Berlin Heidelberg 2007

domain of interest" [7]. We analyse to what extent ontologies formulated in the Web Ontology Language (OWL), are suitable to drive the IE process and how a classical IES has to be augmented in order to deal with such an underlying ontology properly. Within this context we also analyse what kind of management issues regarding the ontology arise and what course of action would be appropriate to dissolve them.

2 Related Work

To use ontologies within the context of 'ordinary' IESs is a relatively new research field. We have to state that to our knowledge only the work of Embley [8] can be considered to be *ontology-driven*, whereas the other works are *ontology-based*, because therein the extraction process is not driven by the ontology rather the ontology is used during extraction as yet another component of the system.

Embley [8] presents an approach for extracting and structuring information from data-rich unstructured documents using extraction ontologies. With data-rich he means data that has a number of identifiable constants such as dates, names, times, and so forth. He proposes the use of the Object-oriented Systems Model (OSM) [9] to represent extraction ontologies, because it allows the formulation of regular expressions as descriptors for constants and context keywords. Both, the generation of the ontology and the generation of the regular expressions require human intervention, because they have to be performed manually. The generated ontology is then parsed to build a database schema that covers the required constants and to generate extraction rules to match constants and keywords. After that, recognisers are invoked that use these extraction rules to identify potential constant data values and context keywords. Finally, the generated database is populated using heuristics to determine which constants populate which records in the database.

Aitken [10] presents an approach to learn IE rules from natural language text using Inductive Logic Programming (ILP). In his approach an ontology is used only as a knowledge bearing artifact that represents the conceptualisation of interest and to which a human annotator can commit to while making annotations on a data corpus. His supervised induction algorithm then uses these annotations to generate extraction rules.

McDowell and Cafarella [11] present an automatic and domain-independent ontology-driven IES called OntoSyphon. Their system takes an ontology as input and uses its content for specifying web searches to identify possible semantic instances, relations, and taxonomic information. For example, for a concept Mammal in an ontology, the system specifies web searches using the phrase patterns introduced by Hearst [12], like "mammals such a", etc. The system then searches the Web for occurrences of these phrases and extracts candidate instances.

However, our aim is to provide an unsupervised ontology-driven IES, that is able to exploit the content of its underlying ontology on its own to extract relevant information from natural language text without being dependent on other knowledge sources.

3 ontoX - An Ontology-Driven IES

In this section, we will explain our method to IE from natural language text, that utilises the knowledge in an ontology. The input ontology is being used as a knowledge bearing artifact that represents the conceptualisation of a domain of interest and the task specification for the extraction task. Our aim by developing this method is to provide a means for 'common' people to perform IE in a way that requires neither skills in particular rule representation languages, nor any other resource but the ontology, such as lexicons, etc. The main architecture of our IES, which we named 'ontoX', is depicted in Fig. 1.

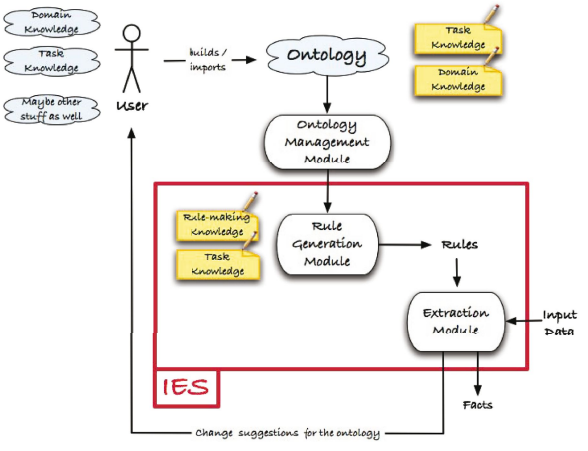


Fig. 1. Main architecture of ontoX

The Ontology Management Module (OMM) takes the input ontology and tries to exploit the knowledge in it to determine what exactly has to be extracted from the input data. Although our method can be applied to any other representation language, our implementation supports ontologies formulated in the Web Ontology Language (OWL) version 1.0 [13] [14]. The Rule Generation Module (RGM) uses the output of the OMM and performs several steps to formulate rules (in our case regular expressions) to locate candidate values that are relevant according to the input ontology. The Extraction Module (EM) takes these rules and determines candidate values in the input texts and applies several heuristics to choose the most accurate values among them. This module finally returns the extracted values and also suggestions to the user regarding possible changes in the ontology.

To support the understandability of our extraction method and its implementation, we explain them by using an example. For that purpose we have chosen the domain of digital cameras, because the domain is widely known and its characteristics are suitable for extraction. Figure 2 shows a possible conceptualisation of the domain containing only relevant aspects of the domain we are interested

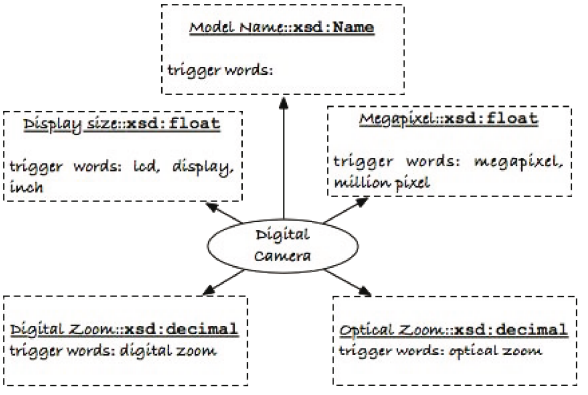


Fig. 2. Graphical representation of our digital camera ontology

in. In this sense, we can think of the ontology as the task specification where the properties of concepts represent the properties for which our extraction method has to determine appropriate values from input data.

3.1 Requirements

Although our method can process any valid ontology that is conform with OWL 1.0, several requirements have to be reconciled by the ontology to enable ontology-driven IE. In this section, we show how these additional requirements can be reconciled by extending ontologies with additional properties. All these properties can be attached to the ontological constructs in OWL using the `owl:AnnotationProperty` element.

Keywords. The most important requirement to enable ontology-driven IE is to enrich the ontological constructs with keywords (i.e., trigger words) that indicate the presence of relevant information in the input text. If the user does not provide a property with corresponding keywords, our system tries to extract appropriate values considering frequent terms in the neighbourhood of other properties' keyword occurrences. Note that this can only return feasible results when enough properties co-exist and not more than one such unknown property has the same data type.

Quality Properties. Within our IE method we allow the user to enrich the constructs in the ontology by a property *confidence_level*, which can take values between $[0, 1]$ indicating the level of confidence of the ontology engineer that this construct is really relevant for the domain. This property helps our extraction method to make decisions when the same value is tried to be assigned to two different properties. In such a case, the property with the higher confidence level would be the winner.

Another property that can be attached to ontological constructs is the *relevance* property that can take one of the two values {true, false}. By marking a construct with this property the user can tell the system that she is not interested in the construct as far as the task specification is concerned, but merely that the construct is part of the domain of interest.

Value Constraint Properties. Constraining properties are required to narrow the search range of possible values of properties that have to be found. In our method, this is possible as far as the OWL 1.0 specification allows it. This means, that our extraction system only takes the XML Schema data types into account whose usage is legal in OWL 1.0.

Temporal Properties. Temporal properties can be useful in the context of IESs to enable two kinds of services. The first one is temporal extraction and the second is change management. With the first one, a user can state that she wants her input data to be extracted using ontological components that are valid at a certain point of time. With the second one, the user can be provided with suggestions regarding out-of-date concepts because they did not appear in the input texts anymore, so that she could adopt the ontology if necessary.

To enable both kinds of services, we suggest the use of *valid_time_begin* and *valid_time_end* properties that can be stated for every construct in the ontology. By doing this, the input data will be analysed using only ontological constructs that are valid at the given point of time, which must be provided by the user of the system in a way that is conform with the `xsd:Date` data type. If no date is given, all constructs in the ontology will be used for extraction.

To enable change management we also provide the user with the property *value_change_frequency* that can take one of the two values {stable, frequent}. The user can mark ontological components with this property in order to indicate that the component will appear in the input text consequently (i.e., stable) or not (i.e., frequent). Using this value an IES can compute accuracy levels for ontological components after each extraction operation, looking whether appropriate values for the component had appeared in the input text or not.

3.2 The Rule Generation Module

The Rule Generation Module (RGM) of ontoX is responsible for generating extraction rules that can be used to identify candidate values for the attribute-value pairs generated by the OMM. Because OWL 1.0 supports only a predefined set of data types, at the present this module merely has to look up appropriate regular expressions for the stated data types.

However, for other ontology representation languages the RGM would have to do a lot more. For the oncoming standard OWL 1.1, which will support user-defined data types for example, the RGM would have to parse the data type definitions in order to generate appropriate extraction rules.

3.3 The Extraction Module

The Extraction Module (EM) of ontoX is responsible for applying the rules generated by the RGM and using them to identify candidate values for properties in the ontology. The steps that are taken by the EM can be considered in two main parts: preprocessing step and extraction step.

Preprocessing. is needed to transform the input data into a format that can be processed more easily by successive modules (e.g., eliminating noisy data that could affect the performance of the system). The preprocessing phase of our extraction method comprises the following particular steps.

Removing Stop Words. Natural language texts are usually loaded with words, called stop-words, which do not convey relevant information but occur frequently in the text as constructs to build grammatically correct sentences. Because of their high frequency, they often cause outliers in term frequency histograms and have to be eliminated from the input text before starting the actual extraction phase. As we evaluate our method with texts in English, at this step we eliminate all stop words commonly used in English.

Locating Data Type Occurrences. We stated earlier that data type properties represent our main interest w.r.t. extraction, for they define the actual values of interest to be extracted. However, the XML Schema data types have different value spaces, which range from integers, bytes, floats to more general data types such as strings. It is clear that the more constraints stated about a property, the easier it will be to identify a particular text as a value of that property. In order to rule out cases where the data types with wider value spaces overlap data types with narrower value spaces, we begin with locating occurrences of the narrower ones.

Values of data types derived from `xsd:string` are much more harder to identify than numbers and other data types that have fixed formats. Therefore, simple pattern matching methods cannot be applied on them. For these data types, especially for the data type `xsd:Name`, we use a heuristic that computes the *named_entity_probability* of a string. To compute this value, we consider anomalies in a string, like as numbers and characters appearing in the same string, or mixed lower and upper case usage of characters that could indicate that the string at hand is not a proper word

Extraction. The extraction phase is the actual phase where the located data type values are going to be assigned to corresponding properties in the ontology. Figure 3 depicts sample input data after the preprocessing phase. As you can see, all the stop words have been removed, all occurrences of data type values that are allowed in OWL 1.0 are underlined, and keyword occurrences are also highlighted by underlining them with two lines (i.e., 'megapixel', 'optical zoom', 'digital zoom', 'inch', 'LCD').

Now we have to assign appropriate values to the properties in the ontology that the identified keywords belong to. For that purpose, we are looking for

5.0-megapixel D-595 Zoom combines power manual control convenience easy-to-use design incredible results. Featuring manual control shutter speed aperture options, 3x optical zoom 19 shooting modes, D-595 Zoom combines manual settings ease of point-and-shoot, packing high-end advanced features affordable, easy-to-use compact camera.

3x optical zoom lens (38-114mm equivalent 35mm photography f2.8 - f4.9) combines 4x digital zoom deliver total 12x zoom so virtually photo opportunity out of reach. super macro mode possible capture amazingly small details, tiny date back penny, from close 0.8 inches.

TruePic TURBO super-charged image processor significantly enhances image quality processing speed. Users experience rapid startup, shutter release playback, capture sharper, more realistic images. One greatest benefits digital photography ability compose review images LCD, allows users accurately frame photographs make on-the-spot decisions about saving deleting them. ensure users can easily view subject frame best-shot possible, D-595 Zoom has, 85,000 pixel 1.8-inch Semi-Transmissive TFT LCD - easy viewing even bright sunlight.

Fig. 3. Sample input data after preprocessing

values that are conform with the predefined data type of the property. In fact, we are looking for the first such value at the left side and the right side of a keyword occurrence, because it is more likely that the values are located near the keywords. For the keyword 'megapixel' that must have a value conform with `xsd:float`, we have the value '5.0' at its left and no appropriate value at its right, because there is a sentence boundary between the keyword and the next valid value. The heuristic used to choose appropriate candidate values for each keyword occurrence is formulated in the following algorithm.

Algorithm 1: Selecting Candidate Values

```
L_k := List of all keyword occurrences in Input
L_d := List of all data type occurrences in Input

for each keyword k in L_k do
 r := k.datatype  // look up the data type of k
 dp := k.property // look up the property to which k belongs

 repeat
   c_l := next token at the left
 until c_l is a valid value or a sentence boundary

 repeat
   c_r := next token at the right
```

```
 until c_r is a valid value or a sentence boundary

 if c_l == sentence boundary and c_r is not, then
   candidate_value = c_r
 else if cr == sentence boundary and c_l is not, then
   candidate_value = c_l
 else   // both, c_l and c_r are valid values
   c_l.evidence = compute evidence for c_l
   c_r.evidence = compute evidence for c_r

   if c_l.evidence > c_r.evidence then
     candidate_value = c_l
   else if c_r.evidence > c_l.evidence then
     candidate_value = c_r
   else   // both values have the same evidence
     if r == xsd:string or a derivative of xsd:string then
       candidate_value = c_r
     else
       candidate_value = c_l
     end if
   end if
 end if

 add candidate_value to the list of candidate values of dp
end for
```

A property may have been provided with more than one keyword, in which case every occurrence of each keyword will be encountered to collect candidate values in the input text. While collecting candidate values, our method marks them with a level of *evidence* that is computed as

$$evidence = \begin{cases} \frac{1}{d} \text{ if } d > 0 \\ 1 \text{ if } d = 0 \end{cases} \quad (1)$$

whereas d is the distance of the candidate value from the keyword occurrence in terms of the words that lie between them. This inverse distance function is used to favour data type values that are near to the keyword over values that are more far away.

If the same data type value appears more than one as a candidate for a certain property, the level of evidence of this value is being updated to the maximum between them:

$$evidence = max(evidence_{old}, evidence_{new}) \quad (2)$$

After having identified all candidate values for a certain property in the ontology we have to choose the final result among them. We already mentioned that OWL allows the definition of functional properties (i.e., `owl:FunctionalProperty`) that

Table 1. Candidate values and the final decisions for properties in the ontology

	Model	**Megapixel**	**Display**	**Optical zoom**	**Digital zoom**
Data type	xsd:Name	xsd:float	xsd:float	xsd:decimal	xsd:decimal
Candidate values	D-595 Zoom Semi-... TFT TruePic TURBO	5.0 1.8	1.8 0.8	3 19 4	4 12
Final decision	D-595 Zoom	5.0	1.8	3	4

is, properties that can have only one value. For these kinds of properties our method chooses the candidate value with the highest computed evidence. For other properties it presents all candidate values whose evidence are above a user defined threshold.

For our sample input text in Figure 3 we would have the attribute-value pairs and their candidate values with decreasing order of their computed evidence as shown in Table 1.

3.4 Change Detection

We think that it is essential to provide ontology-driven systems with means to enable the detection of changes in the underlying conceptualisation. Therefore, we mentioned earlier that ontologies have to be enriched with several additional components, such as the *value_change_frequency*. We think that in our setting it is sufficient to generate a log-file that shows which ontological constructs became out-of-date over time to be suggested to the user for her to change the ontology because it apparently contains constructs that do not occur in the input files anymore.

Our heuristic to detect out-of-date constructs works in a way that incorporates the property *value_change_frequency* and the amount of time over which a property in the ontology did not occur in the input files. The first idea was to decrease the value that indicates the probability that a certain construct in the ontology is still accurate (property *accuracy*) for let's say 20% every time a construct did not appear in the input data. It is clear that such a course of action would not lead us to satisfactory results, because it does not encounter the predicted value change frequency or the confidence level of the construct, which indicated the level of confidence for this construct at the first place. Therefore, we adjusted our idea and defined our function to determine the value of accuracy as follows:

$$accuracy_{new} = accuracy_{old} - \frac{(accuracy_{old}/5) * value_change_frequency}{confidence_level} \quad (3)$$

This function ensures that the value of accuracy decreases at a faster pace for constructs that are said to be 'stable' than for constructs that are predicted

to be 'frequent' anyway. The function also ensures that the value decreases at a slower pace for constructs of which the ontology engineer was more certain.

4 Experimental Results

In this section, we present the evaluation results of our proposed method to extract information from natural language text using ontologies formulated in OWL 1.0. To provide We use *Recall* and *Precision* [15] as established evaluation measures among the community. However, the newness of the field of ontology-driven IE and the fact that there are no freely accessible evaluation data sets that could be used by different IES developers, makes it hard to provide comparative results of different systems.

4.1 Evaluation of Performance

In the first part of our conducted evaluation we focus on the performance of our proposed extraction method that utilises only a relatively small ontology to extract relevant facts from natural language text documents. Further, we evaluate the usage of ontological relations during the extraction process when no keywords are attached to some properties.

We have chosen the domain of digital cameras, because of its popularity nowadays and the fact that its nature can be captured easily by ontologies. We collected a set of 137 digital camera reviews in natural language text from the Web[1] with over 57,000 words[2]. Table 2 depicts the results of our extraction method for the collected data corpus and the ontology as in Figure 2 in terms of the standard evaluation metrics recall and precision.

Analysing the results, we must admit that we had expect better results for the property 'Megapixel' because the keywords for that property are relatively clear and occur consistently in input documents and because its value space is also

Table 2. Evaluation results for the digital camera ontology in Figure 2

	Number of Facts	Correctly Identified Facts	Incorrectly Identified Facts	Recall	Precision
Model	137	110	28	0,79	0,79
Megapixel	137	70	63	0,51	0,52
Optical zoom	124	105	22	0,84	0,82
Digital zoom	13	6	6	0,46	0,46
Display size	113	93	23	0,82	0,80

[1] http://www.steves-digicams.com/ [accessed last in April 2007].

[2] To download the data set and the corresponding result set visit our project page at http://ieg.ifs.tuwien.ac.at/projects/ontox/

narrow. However, we figured that some reviews contained the megapixel information as decimals and sometimes even in letters (e.g., 'five') causing confusion in the extraction process.

On the other hand, the results for the property 'Model' were a pleasant surprise. We did not expect our method to locate appropriate values for this property that well, which was not provided with any keywords. The reason for this is that the model name of a digital camera appears relatively often in the text and therefore falls more often in the neighbourhood of other keywords' occurrences.

The number of incorrectly assigned values for the 'Digital Zoom' property is due to data type occurrences that could not be assigned to their original properties. Better means to state more complex data types can decrease the number of incorrect values, leading to better precision results.

4.2 Evaluation of Scalability and Portability

To evaluate how our method reacts to changes in the task specification, we changed our task specification so that it stated that the system has to extract also the type of storage medium a digital camera supports (e.g., xD-Picture Card, etc.), the kind of power supply it has (e.g., Lithium Ion Battery, etc.), and also information about supported video formats (e.g., MPEG format, etc.). Table 3 contains the results of our extraction method for the same data corpus and the changed ontology in terms of the standard evaluation metrics recall and precision.

The relatively bad results for the property 'Storage' show how the choice of inappropriate keywords for properties can affect the performance of the system. We figured that the keyword 'storage' is used apparently in another context as well, leading to a lot of incorrectly identified values, which was fostered further by the rather large value space that the data type `xsd:Name` is allowing. The same can be said for the large number of incorrectly assigned values for the properties 'Movie Format' and 'Power Source'. The fact that all of them allow values from the `xsd:Name` value space, lead to a lot of candidate values with some of them apparently very close to the keywords.

However, the aim with this phase in our evaluation was to look how difficult it is to make the system extract information for a different task specification. As such a change only requires the change of the input ontology and not the IES

Table 3. Evaluation results for the changed task specification

	Number of Facts	Correctly Identified Facts	Incorrectly Identified Facts	Recall	Precision
Storage	61	15	56	0,25	0,22
Movie Format	56	41	59	0,73	0,41
Power Source	60	26	64	0,43	0,28

itself, we conclude that the scalability and portability of ontology-driven IESs are indeed much better than of regular IESs.

4.3 Evaluation of Change Detection

Another aspect of our method was to enable basic change detection in the extraction interests of the user. We stated earlier the heuristic our method applies to determine the accuracy of properties to identify properties or concepts that do not appear in the input text over a certain amount of time (see Section 3.3). After every extraction operation the user is given a list of all ontological constructs in increasing order of their accuracy level, so that she can change the ontology if necessary.

During our described evaluation phase, our method returned the following accuracy list after the first ten input texts had been processed:

0,129 accuracy of 'Digital zoom'
0,36 accuracy of 'Optical zoom'
0,6 accuracy of 'Megapixel'
1 accuracy of 'Model'
1 accuracy of 'Display size'

Looking at such a list, the user can see that the 'Digital Zoom' property is not well represented in the input texts, indicating either that the property lost its relevance over time (i.e., the domain of interest had changed over time) or that the keywords or the data type of this property are not suitable to extract the desired information. Both cases require the user to look at the input texts and the ontology to figure out which one of these cases is actually true.

5 Conclusion and Future Work

In this work we presented an extraction method that can extract information from natural language text without using any knowledge resource except an input ontology. We thought that such a system would be useful for people with light-weight extraction demands and people who are not familiar with generating all kinds of knowledge resources (e.g., gazetteer lists, extraction rules, etc.) or do not have access to linguistic processing resources (e.g., part-of-speech tagger, etc.) that other state-of-the-art IESs require to yield feasible results.

Because of this focus of our work, we have to live with some limitations, which can be overcome if several requirements on the input ontologies are reconciled:

- The provided keywords have to be chosen carefully, because they are the trigger words using which our method locates candidate values.
- The ontology should contain as much information as possible regarding the constraints on property values. For example, if it is known that values of a particular property can have only values greater than 1, then its datatype should be `xsd:positiveInteger` instead of just `xsd:integer` or other data types with a wider value space.

In short we can say, that the performance of our proposed method highly depends on the quality of the input ontology. If the user can provide the system with an ontology that reconciles the requirements stated above, the results of the system will be as outlined in the experimental results (see Section 4.1).

The system as it is, can be seen as a scalable and portable IES, because to adopt the system to a changed specification or a new domain, only the ontology has to be changed. Some domains require better performing IESs and therefore it is likely that some IES developers will tailor their systems for only a particular domain at hand. But although this would cause a decrease in the portability of the system, it would be easier still to scale than with other approaches to IE.

However, a lot of work has to be done in this research field to convince users that it is indeed beneficial to use ontologies in conjunction with IESs. Possible directions for future work may include the following:

Developing an Extension to OWL for IE. In this paper we proposed several properties that should be used to enrich the existing components of OWL to represent conceptualisations. We think that the more OWL becomes the quasi standard for representing ontologies among the AI community, an extension to OWL for IE could be useful for researchers interested in IE and thus worth the effort. Such an extension to OWL for IE should contain means to represent user-defined data types, quality properties, temporal properties, and linguistic properties.

Utilising Intentional Knowledge for Better Extraction Results. In our proposed method we focused only on the extensional knowledge present in ontologies but merely neglected the intensional knowledge (i.e., instances) if they were not directly related to properties of interest (i.e., `owl:ObjectProperty`). However, intentional knowledge can turn out to be useful for an IES to compare its identified candidate values with existing values of instances to make decisions based on similarity measurement.

References

1. Berners-Lee, T.: Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor. Harper San Francisco (1999)
2. Guarino, N.: Formal ontology and information systems. In: Guarino, N. (ed.) Proceedings of the First International Conference on Formal Ontologies in Information Systems (FOIS), pp. 3–15 (1998)
3. Yildiz, B., Miksch, S.: Motivating ontology-driven information extraction. In: Prasad, A., Madalli, D. (eds.) International Conference on Semantic Web and Digital Libraries. Indian Statistical Institute Platinum Jubilee Conference Series, pp. 45–53 (2007)
4. Riloff, E.: Information extraction as a stepping stone toward story understanding. In: Understanding Language Understanding: Computational Models of Reading, pp. 435–460. MIT Press, Cambridge (1999)

5. Appelt, D.: Introduction to information extraction. AI Communications 12, 161–172 (1999)
6. Yildiz, B., Miksch, S.: Ontology-driven information systems: Challenges and requirements. In: Prasad, A., Madalli, D. (eds.): International Conference on Semantic Web and Digital Libraries. Indian Statistical Institute Platinum Jubilee Conference Series, pp. 35–44 (2007)
7. Gruber, T.: Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies 43, 907–928 (1993)
8. Embley, D.: Toward semantic understanding: An approach based on information extraction ontologies. In: Proceedings of the Fifteenth Conference on Australasian Database Conference, Australian Computer Society, Inc., pp. 3–12 (2004)
9. Embley, D., Kurtz, B., Woodfield, S.: Object-oriented systems analysis: a model-driven approach. Yourdon Press, Upper Saddle River, NJ, USA (1992)
10. Aitken, J.: Learning information extraction rules: An inductive logic programming approach. In: Proceedings of the 15th European Conference on Artificial Intelligence (ECAI'02), pp. 355–359 (2002)
11. McDowell, L., Cafarella, M.: Ontology-driven information extraction with OntoSyphon. In: International Semantic Web Conference, pp. 428–444 (2006)
12. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th International Conference on Computational Linguistics, pp. 539–545 (1992)
13. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: The making of a web ontology language. Journal of Web Semantics 1(1), 7–26 (2003)
14. Grigoris, A., van Harmelen, F.: A Semantic Web Primer. MIT Press, Cambridge (2004)
15. Cowie, J., Lehnert, W.: Information extraction. Communications of the ACM 39(1), 80–91 (1996)

Improving the Efficiency and Efficacy of the K-means Clustering Algorithm Through a New Convergence Condition

Joaquín Pérez O[1], Rodolfo Pazos R[1], Laura Cruz R[2], Gerardo Reyes S[1], Rosy Basave T[1], and Héctor Fraire H[2]

[1] Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET)
[2] Instituto Tecnológico de Ciudad Madero
{jperez,pazos,greyes}@cenidet.edu.mx,
{lcruz,hfraire}@prodigy.net.mx

Abstract. Clustering problems arise in many different applications: machine learning, data mining, knowledge discovery, data compression, vector quantization, pattern recognition and pattern classification. One of the most popular and widely studied clustering methods is K-means. Several improvements to the standard K-means algorithm have been carried out, most of them related to the initial parameter values. In contrast, this article proposes an improvement using a new convergence condition that consists of stopping the execution when a local optimum is found or no more object exchanges among groups can be performed. For assessing the improvement attained, the modified algorithm (Early Stop K-means) was tested on six databases of the UCI repository, and the results were compared against SPSS, Weka and the standard K-means algorithm. Experimentally Early Stop K-means obtained important reductions in the number of iterations and improvements in the solution quality with respect to the other algorithms.

1 Introduction

The problem of object clustering according to its attributes has been widely studied due to its application in areas such as machine learning [4], data mining and knowledge discovery [3, 11], pattern recognition and pattern classification [2]. The aim of clustering is to partition a set of objects which have associated attribute multidimensional vectors into homogeneous groups such that the patterns within each group are similar.

Several non-supervised learning algorithms have been proposed which partition the set of objects into a given number of groups according to an optimization criterion. One of the most popular and widely studied clustering methods is K-means [10]; particularly, since its implementation is relatively simple. However, the K-means algorithm has the following drawbacks:

- The final grouping depends on the initial centroids.
- The convergence to the global optimum is not guaranteed, and for large problem instances it requires a large number of iterations for converging.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 674–682, 2007.
© Springer-Verlag Berlin Heidelberg 2007

A factor that greatly affects the computational cost of the K-means algorithm is the number of iterations that it needs to carry out, since for each iteration it calculates the distance of each object to the groups centroids. In this work a new convergence condition is proposed, which permits to reduce in many cases the number of iterations and improving the grouping quality. The improvement proposed was obtained by looking at K-means from an optimization perspective; i.e., by studying the behavior of the objective function, which in this case is the square error.

In the following section the traditional K-means algorithm is described.

1.1 Description of the Standard K-Means Algorithm

According to the specialized literature [1, 6, 7, 8, 14, 15] four algorithm steps can be identified:

- **Step 1. Initialization.** A set of objects to be partitioned, the number of groups and a centroid for each group are defined. Some implementations of the standard algorithm determine randomly the initial centroids; while some others preprocess the data and determine the centroids through calculations.
- **Step 2. Classification.** For each database object its distance to each of the centroids is calculated, the closest centroid is determined, and the object is incorporated to the group related to this centroid.
- **Step 3. Centroid calculation.** For each group generated in the previous step its centroid is recalculated.
- **Step 4. Convergence condition.** Several convergence conditions have been used from which the most utilized are the following: stopping when reaching a given number of iterations, stopping when there is no exchange of objects among groups, or stopping when the difference among centroids at two consecutive iterations is smaller than a given threshold. If the convergence condition is not satisfied, steps two, three and four of the algorithm are repeated.

2 Related Work

Several improvements have been carried out on the standard K-means algorithm related to several aspects associated to each of the algorithm steps. Roughly, the algorithm consists of four steps:

1. Initialization.
2. Classification.
3. Centroid computation.
4. Convergence condition.

Regarding improvements to the four steps, probably the one that has received the most attention is initialization; in this respect it is worth mentioning the works of Meila [12] and Peña [15]. Concerning the second step several affinity measures for the group elements have been defined; in this case the contributions in [6, 17] stand out. Regarding the third and fourth steps and according to the specialized literature,

no works are reported on the algorithm improvement. In contrast, this paper focuses on the fourth step proposing a new convergence condition.

On the other hand, other authors have made some important contributions regarding the computational implementation of the algorithm; i.e., special data structures and code optimization for algorithm calculations have been proposed, in this respect the following works can be mentioned [7, 9, 14].

At this point it is convenient to stress that, since our approach applies to the fourth step, it is not intended to compete with the other approaches, but rather it may be used to complement them.

3 Proposed Algorithm Improvement

The standard K-means algorithm was implemented, including several code lines for showing at each iteration the square error of each group and the sum of group errors. For testing purposes the *Diabetes* database of the machine learning repository of the Univ. of California Irvine (UCI) [13] was used, and when observing the square error values of the iterations, it was found that in many cases the algorithm went past a local minimum and continued performing iterations which increased the error. Figure 1 depicts the results of a run using the *Diabetes* database. As can be seen in the figure, at iteration 2 a minimum of the square error is attained, which differs from the final value of the square error.

On the other hand, in several experiments the algorithm converged to a local minimum as shown in Figure 2. In these cases the square error followed a decreasing pattern as iterations proceeded.

In order to identify the occurrence rate of this behavior, the algorithm was executed 30 times with the same database (but different initial centroids), and it was found that in 73% of the runs the algorithm went past the local minimum and in 27% of the runs it converged to the local minimum. Encouraged by these results, a new set of experiments was carried out with another five real-world databases from UCI [13], which are briefly described below:

- **Vehicle Silhouette Database.** Its purpose is to classify a given silhouette into one of four types of vehicles, using a set of characteristics extracted from the silhouette. The vehicle can be viewed from many different angles. *Vehicle* contains 846 instances, four groups and 18 attributes.
- **Glass Identification Database.** It is used for classifying types of crystals and it was motivated by criminology investigation. *Glass* contains 214 instances, seven groups and 9 attributes.
- **Diabetes Database.** The *Diabetes* database holds the diagnosis of diabetes patients according to the World Health Organization. Diabetes contains 768 instances, 2 groups and 8 attributes.
- **Heart Disease Dataset**. This database stores heart disease diagnosis. *Heart* contains 270 instances, two groups and 13 attributes.
- **Wine recognition database.** This data consists of the results of a chemical analysis of wines produced in an Italian region and derived from three different vine varieties. *Wine* contains 178 instances, 3 groups and 13 attributes.

- **Liver disorders database.** This data is the result of a medical investigation of patients that can be sensitive to liver disorders, which can be originated by alcohol abuse. *Liver* contains 345 instances, 2 groups and 6 attributes.

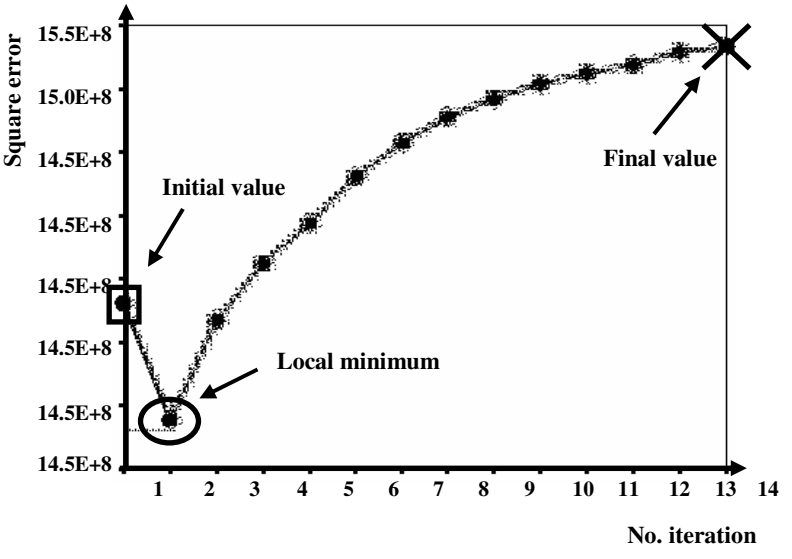


Fig. 1. Convergence to a solution past a local minimum

Table 1. Information of the databases used in the experiments

Database	Number of Groups	Number of Attributes	Number of Instances
Vehicle	4	18	846
Glass	7	9	214
Diabetes	2	8	768
Heart	2	13	270
Wine	3	13	178
Liver	2	6	345

Table 2 shows the results of these experiments, in which the algorithm was executed 30 times for each database.

As can be observed in Table 2, the largest occurrence percentage of the going-past-minimum behavior was 73% and happened for the *Diabetes* database; while the smallest percentage was 20% and occurred for the *Glass* database. This behavior suggested the convenience of defining a new convergence condition for improving the algorithm performance in those cases where the going-past-minimum behavior occurs. Such an improvement can be beneficial in two ways:

- A reduction in the number of iterations (efficiency increase).
- A quality improvement of the solution (efficacy increase).

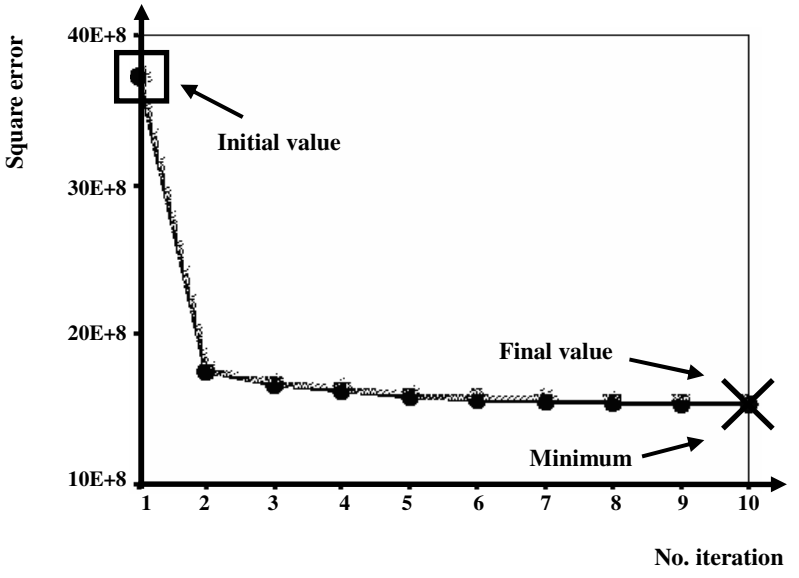


Fig. 2. Convergence to a local minimum

Table 2. Percentage of runs that went past a local minimum

Database	Percentage of runs
Diabetes	73
Liver	70
Vehicle	50
Wine	43
Heart	33
Glass	20

The following section presents a brief description of the modified algorithm that includes the new convergence condition.

3.1 Proposed Improvement for the Standard K-means Algorithm

Traditionally it has been considered that the K-means algorithm is a greedy algorithm; i.e., it can get trapped at a local minimum; however, analyzing the convergence conditions and the experiments conducted, we found that there does not exist a convergence condition that involves the square error and therefore the standard algorithm can not guarantee convergence to a local minimum, as shown by the experiments conducted (Figure 1). In this sense the main contribution of this work consists of associating the square error values to a convergence condition, so as to stop the algorithm when a local minimum is identified. The new convergence condition gets into action when in two consecutive iterations the square error of the last iteration exceeds

that of the preceding iteration. In this paper the standard K-means algorithm that includes this new convergence condition will be referred to as Early Stop K-means.

Notice that the proposed convergence condition is devised in such a way as to find a solution at least as good as that of standard K-means with a number of iterations smaller than or equal to that of standard K-means.

4 Experimental Results

This section presents the results of the experiments conducted for assessing the new convergence condition for the K-means algorithm proposed in this article.

4.1 Algorithm Implementation

The standard K-means and the Early Stop K-means algorithms were coded in C++ Builder, and the experiments were conducted on a computer with the following characteristics: Pentium 4 processor at 2.4 GHz and 512 MB of RAM.

4.2 Other Algorithms for Result Comparison

In order to compare the results of the Early Stop K-means algorithm and the standard K-means algorithm, the Statistical Package for the Social Sciences (SPSS), and the Weka library of Java were used.

SPSS is a commercial software used in business and research. It includes modules for group identification, similarity measures, and classification. The classification module includes an implementation of the K-means algorithm [16].

Weka is a collection of algorithms for machine learning and data mining tasks. It is a Java library and is an open code under GNU. The set of clustering algorithms implemented in Weka includes K-means [5].

It is important to point out that in our comparative experiments we did not include the K-means variations mentioned in Section 2 "Related Work", since our approach applies to the convergence condition; therefore Early Stop K-means does not compete with those K-means variations but rather it can be used to complement them.

4.3 Description of the Experiments

The six real-world databases mentioned in Table 1 were used, and each one was processed using Weka, SPSS, standard K-means and Early Stop K-means. The databases were processed once with Weka and SPSS, since these packages preprocess the input data for determining the initial centroids, which remain unchanged for all the runs. For the standard K-means and the Early Stop K-means, 30 runs were carried out generating randomly the initial centroids for each run.

4.4 Results Related to the Number of Iterations

This section presents the results obtained from the experiments concerning the number of iterations in which the solution is obtained. The first column of Table 3 shows the names of the databases. The second, third and fourth columns show the minimal,

average and maximal values of the number of iterations for the Early Stop K-means; while the fifth, sixth and seventh columns show the corresponding values for the standard K-means. Finally the eighth and ninth columns show the number of iterations for SPSS and Weka.

As shown in Table 3, for the *Diabetes* database the Early Stop K-means algorithm obtained the solution in 3 iterations; while the standard K-means, SPSS and Weka obtained their solutions in 15, 17 and 9 iterations respectively. On the other hand, in the worst case the Early Stop K-means, the standard K-means, SPSS and Weka obtained their solutions in 8, 9, 9 and 14 iterations.

Table 3. Number of iterations for convergence of each algorithm

Database	Early Stop K-means			Standard K-means			SPSS	Weka
	Min	Avg	Max	Min	Avg	Max	Iter.	Iter.
Diabetes	2	3	10	10	15	17	17	9
Liver	2	4	8	3	9	14	6	7
Heart	2	7	14	2	10	14	13	6
Vehicle	2	10	26	2	13	26	19	10
Wine	2	5	13	2	6	13	8	8
Glass	4	8	18	4	9	18	9	14

4.5 Results Related to Solution Quality

This section presents the results obtained from the experiments concerning the quality of the solution. The first column of Table 4 shows the database names, and the second, third and fourth columns show the percentage of quality improvement of the Early Stop K-means algorithm with respect to the standard K-means, SPSS and Weka algorithms respectively.

Table 4. Percentage of quality improvement attained with Early Stop K-means

Database	Standard K-means (Average Values)	SPSS	Weka
Diabetes	7.05	7.05	33.87
Liver	2.45	2.63	8.79
Heart	0.05	0.07	21.91
Vehicle	0.28	6.02	16.01
Wine	0.2	3.58	27.71
Glass	0.01	0.45	10.71

As can be observed in Table 4, the Early Stop K-means increased the solution quality for the *Diabetes* database by 7.5% with respect to the standard K-means, 7.05% with respect to SPSS and 33.8% with respect to Weka. On the other hand, the smallest increase in quality was obtained for the *Glass* database.

5 Conclusions

Several improvements of the K-means algorithm have focused on the initialization of the algorithm [12]; however, one of the factors that affect most its computational cost is the number of iterations performed for reaching convergence.

This work shows that it is possible to improve the standard K-means algorithm using a new convergence condition. During the study of an implementation of the standard algorithm; specifically, analyzing the evolution of the square error at each iteration, we found that in many cases the algorithm did not stop at a local optimum; i.e., despite having reached a local optimum, the algorithm continued and generated a final solution worse than the local optimum. This algorithm behavior was observed in 73% of the runs using the *Diabetes* database.

A detailed analysis of the standard algorithm revealed that it went past the local optimum, because the stopping condition did not take into account the square error. In contrast, this work proposes a new convergence condition that incorporates the square error, which guarantees that the algorithm stops at local optima, thus reducing the number of iterations and improving the solution quality.

For assessing the proposed improvement, a set of real-world databases taken from the UCI machine learning repository was used [13], which were processed using the Early Stop K-means, the standard K-means, the Weka library of Java, and the commercial package SPSS. The experimental results were encouraging; for instance, for the *Diabetes* database the solution was obtained in 3 iterations with the Early Stop K-means; while it took 15, 17 and 9 iterations respectively for the standard K-means, SPSS and Weka. Regarding solution quality, the Early Stop K-means outperformed the standard K-means, SPSS and Weka by 7.05%, 7.05% and 33.87% respectively. Considering all the six databases used in the comparative experiments, Early Stop K-means obtained a reduction in number of iterations and an improvement in the solution quality.

It is important to remark that the proposed improvement technique is not incompatible with other techniques for improving the K-means algorithm, since these apply to the initialization and classification steps of the algorithm. Therefore, the proposed technique can be combined with some other variants of the K-means algorithm, thus contributing to further improve their performance.

Finally, we comment that a future study will focus on determining the compromise between stopping at the first local minimum vs. carrying out more iterations until finding another local or global minimum. Incidentally, preliminary experiments of the K-means algorithm with 30 runs carried out on a real population-based data warehouse of cancer mortality in Mexico show that in 24 cases the objective function converged to a final value without passing past a minimum, and in 6 cases the objective function converged passing past one or more minima. Of these last cases, in 3 cases the final value was better than that of the first intermediate minimum, and the difference in values was not greater than 0.6%.

Acknowledgement. We express our gratitude to Jesús Salinas C. and Adriana Mexicano S. (students of the MSc program at CENIDET) for their assistance in the algorithm coding and experimentation.

References

1. Bottou, L., Bengio, Y.: Convergence Properties of the K-means Algorithms. Advances in Neural Information Processing. Systems. MIT Press, Cambridge (1995)
2. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley & Sons, New York (1973)
3. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R.: Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press (1996)
4. Fisher, D.: Knowlwdge Acquisition via Incremental Conceptual Clustering. Machine Learning 2(2), 139–172 (1987)
5. Garner, S.: Weka: The Waikato Environment for Knowledge Analysis. In: Proc. New Zealand Computer Science Research Students Conference, pp. 57–64 (1995)
6. Hamerly, G., Elkan, C.: Alternatives to the K-means Algorithm that Find Better Clusterings. In: Proc. 11th International Conf. on Information and Knowledge Management CIKM'02. ACM. Virginia, USA (2002)
7. Kanungo, T., Netanyahu, N.S., Wu, A.Y.: An Efficient K-means Clustering Algorithm: Analysis and Implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7) (2002)
8. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: A Local Search Approximation Algorithm for k-Means Clustering. In: Proc. 18th Annual ACM Symposium on Computational Geometry (SoCG'02). Barcelona, Spain, pp. 10–18 (2002)
9. Likas, A., Vlassis, N., Verbeek, J.J.: The Global K-means Clustering Algorithm. Pattern Recognition, 451–461 (2003)
10. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. Proc. 5th Berkeley Symp. Math. Statistics and Probability 1, 281–297 (1967)
11. Mehmed, K.: Data Mining: Concepts, Models, Methods, and Algorithms. John Wiley & Sons, Chichester (2003)
12. Meila, M., Heckerman, D.: An Experimental Comparison of Several Clustering and Initialization Methods. In: Proc. 14th Conf. on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco
13. Merz, C., Murphy, P., Aha, D.: UCI Repository of Machine Learning Databases. Department of Information and Computer Science, University of California.
14. http://www.ics.uci.edu/~mlearn/MLRepository.html
15. Pelleg, D., Moore, A.: X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In: Proc. 17th International Conf. on Machine Learning (2000)
16. Peña, J.M., Lozano, J.A., Larrañaga, P.: An Empirical Comparison of Four Initialization Methods for the K-Means Algorithm. Dept. of Computer Science and Artificial Intelligence, University of the Basque Country, San Sebastian, España.
17. SPSS, Inc. Headquarters, Chicago, Illinois, http://www.spss.com/es/
18. Su, M.C., Chou, C.H.: A Modified Version of the K-Means Algorithm with a Distance Based on Cluster Symmetry. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(6), 674–680 (2001)

Modelling Agent Strategies in Simulated Market Using Iterated Prisoner's Dilemma

Raymond Chiong

School of IT & Multimedia, Swinburne University of Technology (Sarawak Campus)
Kompleks Negeri, Jalan Simpang Tiga, 93576 Kuching, Sarawak, Malaysia
rchiong@swinburne.edu.my

Abstract. Computer-based simulations have been used extensively to model various economic problems in recent years. However, many of these studies are based on quantitative data that were taken at a certain point of time and thus could be deductive and inappropriate. This paper presents a unique agent based approach that places lower demand on data using Prisoner's Dilemma (PD), a classical non-zero sum game, to model the complexity within a market environment. We create a model with agents acting as firms to perform transactions among one another with chosen iterated PD (IPD) strategies. Our model shows that cooperation could emerge even though the simulated environment is highly variegated.

Keywords: Iterated prisoner's dilemma, agent strategies, market simulation.

1 Introduction

The economic world is essentially a large multi-agent system. Computer-based simulations are therefore a useful test bed for developing real world economic models to investigate the dynamic and complex behaviour of such system. As the real world market is normally driven by numerous players going about their business and interacting with each other through financial transactions, computer models that do not follow the same principles are less likely to be successful than those based on agents interacting with each other. Nevertheless, the agent based approach and their connectivity through market transactions are somehow not sufficient to create a realistic model. The missing ingredient is the strategy that the agent uses when deciding whether or not to perform a transaction.

Much of strategy research have been based on Prisoner's Dilemma (PD), an elegant model of game theory for studying decision making and self-interest. The existing works on PD have shown that a number of different strategies are possible and that they all lead to known but different outcomes. A particularly interesting version of PD occurs when it is played more than once between a range of agents, as it allows for dynamic development and adjustment of strategies by each individual agent. This iterated PD (IPD) is therefore particularly suitable for investigation of the dynamics of the cooperative and competitive behaviour in an economic environment.

Although the IPD has been studied extensively over the past two decades, most of its literature on economics to date focus mainly on measurement techniques, data

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 683–696, 2007.
© Springer-Verlag Berlin Heidelberg 2007

collection methodology, prediction, empirical findings, etc. [1]. According to [2], quantitative data that are used normally come from measurements that were taken at a certain point of time, which could be deductive and inappropriate. As such, the main objective of this paper is to present a unique agent based approach which places lower demand on data using IPD [3-4] to model the complexity of the economy market.

The rest of this paper is organised as follows: the next section introduces the background of PD. Following which, the well-known IPD tournaments and an analysis on different kinds of strategies proposed for IPD throughout its history are presented. Ensuing sections describe some applications of PD to economic problems, the economy model we use and a market competition simulation. Finally, we draw conclusions and highlight some potential future works.

2 The Prisoner's Dilemma

The classical PD is a non-zero-sum game that has been studied widely since its inception in 1950s. It presents a story of two prisoners being interrogated by police in isolation from each other. A confession from any one of the two prisoners and silence from the other would result in a heavy sentence for the latter whilst the former would walk free. However, if both prisoners remain silent, they would only get a light sentence. Under the circumstance whereby both confess, they would receive a sentence slightly less severe than the heavy one.

From the above scenario we see a dilemma among the two prisoners on making their choices on whether to confess or remain silent. It is obvious that the rational choice from the viewpoint of each prisoner would be to confess, because one would be freed if the other remains silent. However, should both of them decide to confess, they are worse off than if they remain silent.

The above scenario is one-off, as the two prisoners make their choice against each other only once. A renewed interest in the PD research in 1980s has inspired political scientist Robert Axelrod to extend the one-off game into a repeated form, which he called the IPD game [5]. To formalise the game, we could visualise two players playing simultaneously against each other with the choice of either to cooperate (remain silent) or defect (confess). The notion of four different payoffs resulting from the players' choices has been adopted by Axelrod [6-7], namely P (punishment), R (reward), S (sucker's punishment) and T (tempting reward). A reward is given when both players choose to cooperate, whereas punishment will be given if both of them choose to defect. In the situation where one player defects and another player cooperates, the one who defects is awarded a tempting reward but the one who cooperates will be given the sucker's punishment. Table 1 summarises the payoffs of the game.

According to Axelrod [5], a dilemma will always exist given the rules $T > R > P > S$ and $2R > T + S$. Based on these two rules, temptation to defect is always greater than the reward to cooperate since $T > R$. Highest individual score of T points will be given to the defector and the lowest score of S points to the cooperator. However, $T + S$ is never greater than $2R$, which means the total score for a group to cooperate is always greater than the total score for an individual to cooperate and another individual to defect. It could be even more unfortunate for both players to receive the lowest total payoff of P points if both choose to defect.

Table 1. The payoff matrix for two-player IPD game

		Player 2	
		Cooperate	Defect
Player 1	Cooperate	*R=3, R=3*	*S=0, T=5*
	Defect	*T=5, S=0*	*P=1, P=1*

3 Axelrod's Computational Tournaments

When the PD game becomes iterated, the players are allowed to form strategies to play the game based on previous moves instead of using mere rationality. The computer-based approach, including the agent based approach, has been used successfully to simulate various strategies in the game. Axelrod pioneered the attempts in computer simulations of the IPD game by conducting two well-known IPD tournaments in early 1980s [5-7]. He invited many experts from diverse areas to create and submit strategies to the tournaments.

It is necessary to note that Axelrod had enforced some rules that all the participants must understand and consider before submitting their strategies to the tournaments. Generally, each strategy must decide its moves, which are composed solely with the combination of two actions – cooperate and defect, without knowing the opponent's action until the game takes place. As the only information available in deciding whether to cooperate or defect is the previous moves of other strategies in the game, each strategy has to remember the entire history of the game.

In the first tournament, there were only 14 strategies submitted by game theorists from diverse fields in economics, biology, political science, mathematics, psychology and sociology [6]. Most people produced their strategies by capturing some behavioural assumptions of the sequence of moves for each strategy and specifying them accordingly to generate the current moves. All the strategies were arranged in a round robin format, competing against one another, against itself, and also against an additional Random strategy that randomises its moves. The calculation of the scores for each strategy was based on the payoff shown in Table 1 above, where the outcome for a particular pair of strategies pitted together was constant.

Based on the payoff structure, all the 14 strategies submitted and a Random strategy were pitted against each other and itself for 200 rounds. The tournament was repeated for five times, and Axelrod confirmed that the winner was a simple strategy submitted by Anatol Rapoport called Tit-for-Tat (TFT). TFT scored an average of 504 points to win the tournament with simply cooperating on the first move and thereafter repeating the preceding move of the other strategy. In overall, the tournament concluded that cooperative strategies had done better than those non-cooperative strategies, as eight of the top scoring strategies were those called by Axelrod as "nice" strategies which have never defected on their first move.

Subsequently, a second tournament was carried out by Axelrod after circulating the results of the first tournament publicly. In the second tournament, 62 strategies were

submitted from six countries [7], where the participants were more widely ranged from computer hobbyists to professors of various expertises. Instead of playing 200 rounds for each strategy as in the first tournament, this time the game was randomly terminated. As a result, an average of 151 rounds was played for each strategy.

Different tournament, but with the same old story – TFT was again submitted by the same person, and again won the tournament in style. All the "nice" strategies were again fared better, except for one strategy called Tit-For-Two-Tat (TFTT). TFTT keeps cooperating unless the other strategy defected twice consecutively. It did well in the first tournament, but did poorly in the second. A thought that comes to mind is that – be nice, but not too nice (not until opponent defected twice)! Axelrod explained that some participants might have taken into account the lessons learned from the first tournament, and thus capitalised on the weaknesses of some strategies in order to exploit them.

Due to the lack of robustness and dynamics in the simulation of his two tournaments, Axelrod later decided to explore further the evolution of cooperation in IPD using Genetic Algorithm (GA). The purpose was to variegate the simulated environment and allow new strategies to be discovered based on the genetic operators of GA such as crossover and mutation [8]. With this new way of implementation, Axelrod again conducted a computational simulation in 1987. In contrast to his previous works, Axelrod imposed fewer restrictions to the game strategies. He considered a set of strategies that were deterministic, where each strategy's decision on its current move was based on the outcome of its previous three moves rather than the entire history.

In this simulation, 20 random strategies had been included, plus another 8 best performing strategies selected from his second tournament. At the conclusion of the simulation, Axelrod found that GA evolved most of the ordinary strategies. However, he specifically highlighted that TFT was one of the strategies that were able to match up with any other strategies in terms of the accumulated payoffs.

Axelrod then removed the 8 specially selected strategies and repeated the simulation with only the 20 random strategies. He noticed that the level of complexity in the simulation intensified due to the concurrent evolution nature of the strategies. The average score accumulated by these evolving strategies gradually increased as the game repeated, hence implied that the strategies were slowly able to differentiate between the cooperators and the defectors.

Axelrod's contribution to the research of IPD has been significant, and the results from his tournaments have been considered by many as a definitive solution to the problem of cooperation in the PD game. In our work, we use Axelrod's research as the framework but build our experiments based on an agent-based economy model we previously developed [9]. We believe that only by imposing the IPD strategies to an environment with real-world ingredients, where the agents are selfish and the resources are limited, the nature of IPD can be fully reflected.

4 Analysis of Strategies

In the early days, the only rational and best strategy for the classical PD was simply to defect all the time, as it was the safest regardless of what action the opponent took. As

explained in the previous section, various kinds of strategies only started to emerge after the tournaments organised by Axelrod in the 1980s with an aim to study strategies that would encourage behaviour of cooperation in IPD. When Axelrod initiated his first IPD tournament, only around 14 strategies had been proposed. Today, there are numerous strategies used in the IPD. Among the most common are All-Cooperate (All-C), All-Defect (All-D), TFT, Spiteful, Pavlov, TF2T and Random. There are also some strategies which are less common, but had been used in several prominent experiments of the IPD in the past, such as soft_majo, per_ddc, per_ccd, mistrust, per_cd, hard_tft, slow_tft and hard_majo.

TFT, a very simple strategy and overall winner of all Axelrod's IPD tournaments, has established its reputation among all other strategies as the most robust strategy available in PD game on both experimental and theoretical grounds (see [5-7]). TFT is simple in the sense that it cooperates on its first move and follows opponent's moves subsequently. According to Axelrod, it has three characteristics that account for its impressive performance:

1. Nice (cooperates on the first move).
2. Retaliatory (punishes defection in the prior move with defection).
3. Forgiving (immediate return to cooperation after the adversary cooperates).

Although TFT has been claimed as the best strategy in IPD for the past twenty years since Axelrod's tournaments, a lot of game theorists in this field have never ceased to depose its status with endless research to investigate more winning strategies. Some researchers even protested that logically speaking TFT can only come to a draw scenario, and it can never win or score more points than its opponent in a single game since its first move is always to cooperate [10].

In early 1990s, Nowak and Sigmund [11] studied the robustness of TFT in heterogeneous populations, which brought the game into a new dimension with the introduction of noisy environments. They mirrored the noisy environments to real-life situations, and concluded that TFT was prone to making mistakes and did not endure in such environments. Instead, they promoted another strategy that exhibits Win-Stay, Lose-Shift characteristic called Pavlov in their subsequent research [12]. Pavlov cooperates on its first move and then cooperates again only if a previous move is a symmetrical move: either both strategies cooperate or both defect. In summary, Nowak and Sigmund considered Pavlov to have four advantageous attributes from other strategies:

1. It is nice and is able to maintain cooperation.
2. It punishes defection with no hesitation.
3. It restores mutual cooperation but is not exploitable, as it starts to defect if it fails to convey its objective to cooperate to its opponent.
4. It tends to exploit universal cooperation.

Though Pavlov has the ability to recover from mistakes as we can see in the third attribute above, Nowak and Sigmund resigned to the fact that Pavlov fails to All-D, a strategy that defects on every move regardless of what other does, due to its objective to maintain cooperation whenever it is possible.

Later in the mid 1990s, Beaufils, Delahaye and Mathieu [13] created a new strategy called Gradual, which they claimed to outperform TFT. They carried out

their experiments with the same approach based on the tournaments of Axelrod, and found that Gradual shares all the good qualities of TFT but by a step further, it is not as simple as TFT. Gradual generally cooperates on its first move, while retaliates upon defection on the subsequent moves. On its opponent's first defection, it reacts in a unique way with a defection and two cooperative moves. On the opponent's second defection, it will defect continuously for two times followed with two cooperative moves. On the opponent's third defection and so forth, it reacts accordingly with subsequent number of defections but always followed with two cooperations after its defections. Based on this, Beaufils et al. derived four principles of Gradual from their experiments:

1. Do not be envious.
2. Do not be the first to defect.
3. Reciprocate to not just cooperation, but also defection.
4. Do not be too smart.

As we move into the 21st century, a 20th Anniversary IPD Competition has been organised in 2004 based on the original Axelrod's tournaments. The competition attracted 223 entries, and through a round robin setup, a new strategy submitted by a team from University of Southampton in England proved to be the most successful. This strategy makes a series of moves using agents to recognise each other and act cooperatively. The University submitted 60 strategies to the competition, which were designed to recognise each other through a series of five to ten moves at the start. Once this recognition was made, one strategy would always cooperate and the other would always defect, assuring the maximum number of points for the defector. If the strategy realised that it was playing a non-Southampton strategy, it would continuously defect in an attempt to minimise the score of the competing strategy. As a result, while some strategies from the Southampton team had taken up the top spots in the competition, there were a number of strategies from the team occupied positions towards the bottom.

Although this strategy is notable in that it proved more effective than TFT, it takes advantage of the fact that multiple entries were allowed in this particular competition. In a competition where one has control of only a single player, TFT is certainly a better strategy. It also relies on circumventing rules about the classical PD in that there is no communication allowed between the two players. When the Southampton strategies engage in an opening "ten moves" to recognise one another, this only reinforces just how valuable communication can be in shifting the balance of the game.

5 Applications of Prisoner's Dilemma to Economic Problems

Following Axelrod's ground-breaking research, some excellent work has been carried out in the simulations of markets and IPD. One such example was a new complex tournament staged by Rust, Palmer and Miller [14] that simulated a double-auction market, where the winning entries tended to be simple and contained little intelligence. In this section we discuss only a few papers which could have ample contribution to our experiments.

Economics related PD works have substantially focused on how cooperation may be maintained or disturbed when the game is repeated [15]. In a most common setting, the game in an economic context is usually played with pairs of agents facing one another repeatedly based on a chosen strategy, or in a spatial form, a group of agents adopting different strategies plays against their direct neighbours. Lindgren [16] studied the interaction among agents who made decisions based on a set of strategies within an economic environment. In his work, agents acquired finite memory and evolved over time with fairly simple strategies such as cooperate, defect and TFT. He observed that in a finitely repeated game, agents made mistakes from time to time but possessed a self-organising ability to stay in the game. His results showed that strategies were constantly changing and rarely settling down. Therefore, there was no equilibrium in the game.

Another typical work in the same context but with different background can be found in Arthur et al. [17]. Arthur and his colleagues studied an artificial stock market model developed at Santa Fe Institute. The stock market model was a multi-agent model with numerous market predictors. In their model, agents adopted a random set of expectations rather than strategies, and search among the market predictors. The agents would either adopt or discard the predictors based on their accuracy. Updating process was taking place over time, with approximately 20% of the poorly performed predictors being replaced. The results of this work showed that there were two competing outcomes – the set of expectations adopted by agents evolved either into a stable or unstable way. We see that if the model were given any higher intensity of choice, variety of complex dynamics can emerge.

The oligopoly game is yet another good example of economic problem studied using the PD game [18]. According to Friedman [19], oligopoly game focused on the strategic behaviour among small number of competitors in oligopolistic markets. In Friedman's oligopoly game, a number of firms were offering homogenous products to a market, and the price a single firm set to sell its product would affect other firms' market share. He demonstrated the logic of self-interest behind non-cooperative actions in a one-off game, but showed that cooperation was likely to spread when the game was simulated in a repeated form with unknown length due the possibility of punishing non-cooperative actions in later rounds. In conclusion, he indicated that firms were likely to go through an adaptive learning process in setting their prices by observing the behaviour of other firms during the build up.

6 The Economy Model

The main goal of our work is to examine the usefulness of IPD strategies in economic context and investigate the dynamics of behaviour that emerges through the interaction of those strategies. In doing so, we adopt an economy model [9] that simulates a market environment. The economy model uses bit strings to represent natural resources and products created from those natural resources, as described by Kauffman [20]. Each firm in the simulated market can produce its own products from the resources it has and can exchange the products – or even the resources in its original form – with other firms during the trading process.

The firms are modelled as autonomous agents with internally stored state information. The general internal structure of the agent is shown in Figure 1. During the trading process, each firm can either be a buyer or a dealer as the situation in the market warrants. This means that the firms are allowed to trade freely with any other firms without any restriction and from time to time the firms can switch from buying to selling or vice versa.

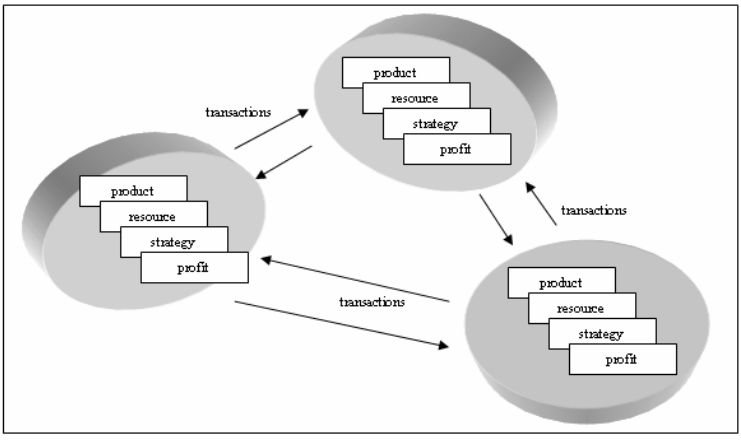


Fig. 1. Internal Structure of Agents

In the simulated market, it is assumed that there is a binding agreement between firms to a standard market price during the trading process. Therefore, each firm has the choice of selling its products to other firms either at a market price (cooperating) or disregarding the agreement by selling at a higher price (defecting). Although selling products at a high price allows a firm to gain more profit in one single trade, it is not necessary that it will yield higher total profit at the conclusion of the entire trading process. This is because some firms that adopt reciprocal strategies will tend to defect in return. When both firms that are involved in a trade interaction decide to defect, the trade between the two firms is halted immediately and thus no profit is distributed between them.

As the aim of each firm is to maximise its own profit, every firm is repeatedly engaged in the market activities in search for the best available strategy that will increase their total profits at the conclusion of each trade cycle. The underlying idea of the simulation is that over time, firms that are unsuccessful tend to look around and probably to imitate strategies of other firms that appeared to be more successful. In other words, unsuccessful firms are likely to copy the strategy of the more successful firms they used to interact with. Therefore when the simulated environment changes, the more successful a firm is, the more likely its strategy is to be copied or imitated by other firms, where the measure of success is simply the profits generated by each firm based on its current strategy. It is necessary to note that this process is not deterministically induced by any mathematical structure of the model or explicitly programmed, but purely based on the strategic interaction among the firms.

6.1 Implementation of the Model

As aforementioned, our economy model uses bit strings to represent natural resources and products created from those natural resources. Agents, which represent firms, operate on these strings using grammar rules. A grammar rule consists of an input/output pair of bit strings. When a grammar rule is applied to a given bit string, if the input string is contained within the given string as a substring it will be replaced with the output string. Only one instance of the input string within the given string, which is chosen at random, is replaced if the given string contains more than one copy of the input string.

Each firm, or agent, in the simulated market can produce its own products from the resources it has and exchange the products or even the resources in its original form with other firms during the trading process. As such, the economy model maintains two 'pools' of bit strings, the resource pool and the product pool. The resource pool contains a set of randomly generated initial strings which are always available. When a grammar rule acts on a resource string to change it into a new string, the original resource string is still preserved. The product pool is initially empty, and any new strings created from resource strings are placed in it. When a grammar rule acts on a product string, the original is removed and replaced with the new string. Thus resource strings in our model are in infinite supply but product strings are not.

Activities in the simulated model are monitored in a sequence of generations. With a 0.5 probability, either a resource string or a product string is selected at random at each generation. The selected string is passed to an agent, which is also chosen at random. The agent selects one of its grammar rules, and applies it to the selected string. If the grammar rule matches, the string is transformed.

If the selected string was a resource string, it remains in the string pool when the transformed string is added. If the selected string was a product string, however, the original string is removed from the string pool. This makes all the initial resource strings always available in the pool.

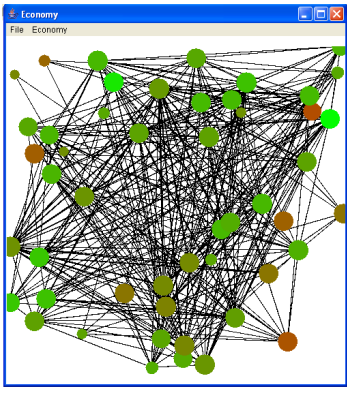


Fig. 2. Graphical representation of the simulated market

The model also adopts a simple graphical user interface (see Figure 2) that helps to visualise the outcomes of market activities during the trading process through

real-time animation. The animation allows us to see the simulation dynamics while the agents interact within the market. In addition, an output screen that displays the simulation results is used to report the performance of agents in a graphical form. The graph measures the accumulated profit of each agent during the trading process. The results at the end of the simulation can be saved and read from a text file.

The image in Figure 2 shows the environment of a simulated market where agents are represented as circles. The colour of a firm can vary from brown to dark green. If an agent consumes more resources than products, its colour is brown. As the agent starts to produce more products from the resources it has, its colour changes proportionally from brown to dark green. The size of the agent, which is determined by the radius of the circles, represents how actively an agent is engaged in the market activities. The size increases if the agent is active in its trading but not necessarily representing higher profit. Over time, the size decreases if the agent does not manage to trade its products or resources with other agents for profits, thus signifying that the products or resources have expired after certain period of no trade. Trade interaction between two agents is represented by drawing a line between them. The thickness of the line is proportional to the intensity of trade interaction between corresponding agents. The thicker the line is, the more frequent the trade of products between the two agents.

6.2 Strategy Representation

As for the agent strategies, we adopt the representation developed by Axelrod [8]. Under this representation, bit strings of 0s and 1s are used to encode different strategies. A 0 is corresponding to a cooperation and 1 to a defection. An agent in the game has to keep the history of the past three rounds of play to decide on a current move. Each of these three rounds is recorded as a bit pair, thus a 6-bit string is produced for a three-round memory. For instance, 11 11 11 means both agents defected for the past three rounds, 00 00 00 means both agents cooperated for the past three rounds, 01 01 01 means the first agent cooperated while the second agent defected for the past three rounds, and so on. In deciding on a current move, a strategy must contain all the possible actions for the previous three rounds, resulting in $2^6 = 64$ possibilities, and thus a 64-bit string is required. Since there is no memory of previous rounds at the beginning of the game, an additional 6-bit is required to be the initial memory. This culminated in a strategy being encoded with the representation of a 70-bit string.

After devising a way to represent the game strategies, an initial population of agents with different strategies needs to be determined. It is almost impossible for us to include all the game strategies from Axelrod's tournaments in our simulation experiments, therefore only six distinctive strategies are chosen, namely All-D, Random, Prober, Mistrust, TFT and All-C. In addition, Pavlov and Gradual, the two outstanding strategies claimed to be better than TFT, are also included. We are unable to include the winning strategy from Southampton team at the 20th Anniversary Competition as it is more like a collusion program than a formal strategy.

We incorporate these eight IPD strategies into the model we have just described. Every agent thus has a choice of two actions to either defect or cooperate during each interaction based on the selection of strategies. Various colours are used to represent

different strategies selected by different agents. The colours replace the original colours of the agents based on the strategy each agent has chosen. When an agent decides to adopt another strategy from a more successful agent in order to increase its profit during a trading interaction, the colour of the agent will change, thus reflecting the new strategy of its choice. Table 2 below shows the representation of colours for different strategies.

Table 2. Strategy representation with different colours

Colours	Strategies
Red	All-D
Green	Random
Yellow	Pavlov
Blue	Prober
Magenta	Gradual
Cyan	Mistrust
Gray	TFT
Black	All-C

7 Market Competition Simulation

In this section, we present our simulation experiments with our economy model to examine the performance of the eight selected strategies dynamically in order to determine the winning strategy in a market competition. Each agent is allowed to trade freely with any other agents over 100 generations with the aim to achieve maximum profit. An agent excels above other agents when it acquires a higher profit. We monitor the interactions among firms throughout the trading process and compare the total profits made by all the firms to eventually determine the most profitable strategy within the simulated market. During the trade interactions, all agents are able to learn and improve their profits by either maintaining their current profit-making strategies or shift to adopt the more profitable strategies from among its counterparts. Figure 3 shows the initial distribution of firms with different strategies.

After a period of simulation run, the distribution of strategies changed from one generation to another. We noticed that the less cooperative strategies had a higher distribution rate at the initial stage as compared to the more cooperative strategies. However, once those more cooperative firms that have adopted reciprocal strategies such as TFT and Pavlov started to punish against defection, firms who have adopted less cooperative strategies began to realise that their strategies were not going to bring them more profits. Thus, to ensure their continual survival within the market, more and more firms gradually decided to forego their less cooperative or selfish strategies and started to work more cooperatively. We observed this scenario after 15 generations. The simulation over 100 generations had displayed a very interesting pattern, as shown in Figures 4 through 6.

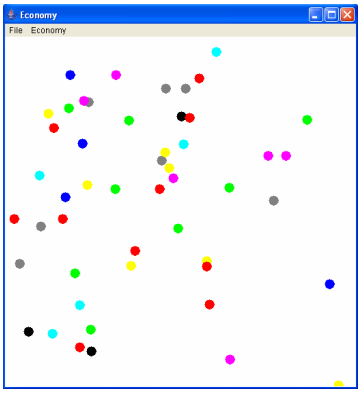


Fig. 3. Initial distribution of firms with different strategies

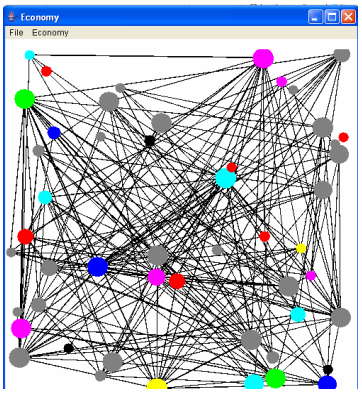


Fig. 4. Distribution of strategies after 25 generations

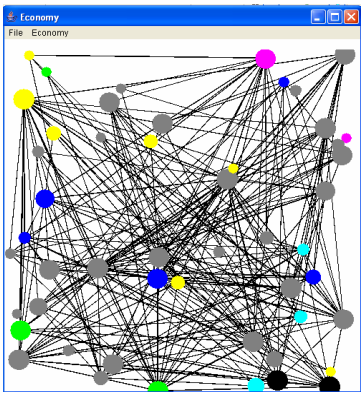


Fig. 5. Distribution of strategies after 50 generations

Fig. 6. Distribution of strategies after 90 generations

We repeated the same simulation for another five times. From these subsequent simulations, TFT has shown to be robust and excel under different situations, except on two occasions. In these two occasions, All-C and Prober emerged to be the respective dominant strategies.

In the simulation when All-C became the dominant strategy, the initial distribution of the strategies showed that 11 out of the 50 firms had adopted TFT, which was more than 1/5 of the overall population. We believe that the popularity of TFT strategies at the initial stage caused an early extinction of defecting strategies such as All-D or Mistrust, while All-C was able to maintain a fairly constant representation and eventually overturn TFT to become the overall winning strategy.

In another simulation, Prober, a fairly unpopular strategy that had performed so badly in previous simulations, sprung a big surprise to emerge as the dominant strategy at the conclusion of the 100 generations. This result showed us that a strategy that performs poorly against non-cooperative strategies might have the potential to

excel while competing against the more cooperative strategies. We observed that the poor performance of TFT had encouraged the emergence of Prober to become the winning strategy. Under these circumstances, TFT's failure had also created a negative impact on the success of All-C.

The results from our experiments showed that TFT proved to be robust in a new way, though the emerging patterns of the simulated market varied from simulation to simulation. In fact, the behaviour of our model is equally unpredictable as to the real-world economy. We have seen how each strategy reacts to different market environment, and how each strategy is affected by other strategies. We observed that whenever the profit declines, the market becomes more cooperative.

8 Conclusion and Future Works

In this paper, we investigated the application of IPD game to economic modelling and provided an experimental evidence to demonstrate the usefulness of IPD in simulating the dynamics of market behaviour in an economic environment. Our work showed that cooperation is the optimal solution in economics, and that a winning strategy must exhibit three qualities: (1) the ability to cooperate with cooperating strategies; (2) the ability to reciprocate against defecting strategies; and (3) the ability to exploit cooperating strategies during mutual cooperation.

We believe that our study could provide an alternative way of thinking about certain aspects of economics and learning how to understand some real and practical economic problems. Though our work cannot predict the outcomes of many real-world transaction scenarios, it enables us to model a certain market condition and envisage what this condition will grow into after a set period of time.

Before we end, we would like to stress that our experimental model and methodology are not without limitations. In fact, there are still many possibilities for our model to be expanded further. One possible extension is to use GA to evolve new population of agent strategies from the existing population. GA could be very useful for optimising the outcome of the economy market. A preliminary result can be found in [21], where different parameters of GA had been investigated for the purpose of fine-tuning the market condition. More information on the possible uses of GA within economics can be found in the work by Sargent [22].

Finally, we would like to reiterate that complexity and uncertainty are closely related to great part of economic problems. An analytical solution of these problems may be hard or even impossible to obtain. This paper presented an artificial adaptive agent approach which we believe can provide more freedom for testing hypotheses and give powerful insights into strategic behaviour of firms and optimal business strategies, thus offering more flexibility and ability to experiment on issues from the real world.

References

1. Paldam, M., Svendsen, G.T.: Missing Social Capital and the Transition in Eastern Europe. Journal of Institutional Innovation, Development and Transition 5, 21–34 (2002)
2. Gilbert, N.: Agent-based Social Simulation: Dealing with Complexity. Retrieved from http://www.complexityscience.org/NoE/ABSS-dealing%20with%20complexity-1-1.pdf

3. Poundstone, W.: Prisoner's Dilemma. Doubleday, New York (1992)
4. Darwen, P., Yao, X.: On Evolving Robust Strategies for Iterated Prisoner's Dilemma. In: Yao, X. (ed.) Progress in Evolutionary Computation. LNCS, vol. 956, pp. 276–292. Springer, Heidelberg (1995)
5. Axelrod, R.: The Evolution of Cooperation. Basic Books, New York (1984)
6. Axelrod, R.: Effective Choice in the Prisoner's Dilemma. Journal of Conflict Resolution 24, 3–25 (1980)
7. Axelrod, R.: More Effective Choice in the Prisoner's Dilemma. Journal of Conflict Resolution 24, 379–403 (1980)
8. Axelrod, R.: The Evolution of Strategies in the Iterated Prisoner's Dilemma. In: Davis, L. (ed.) Genetic Algorithms and Simulated Annealing, Morgan Kaufmann, San Francisco (1987)
9. Chiong, R., Wong, D.M.L., Jankovic, L.: Agent-based Economic Modelling with Iterated Prisoner's Dilemma. In: Proceedings of the IEEE International Conference on Computing and Informatics (ICOCI). Universiti Utara Malaysia, Kuala Lumpur, Malaysia (2006)
10. Sober, E., Wilson, D.S.: Unto Others: The Evolution and Psychology of Unselfish Behavior. Harvard University Press, Cambridge (1998)
11. Nowak, M.A., Sigmund, K.: Tit-for-Tat in Heterogeneous Populations. Nature 355, 250–252 (1992)
12. Nowak, M.A., Sigmund, K.: A Strategy for Win-Stay, Lose-Shift that outperforms Tit-for-Tat in the Prisoner's Dilemma game. Nature 364, 56–58 (1993)
13. Beaufils, B., Delahaye, J.P., Mathieu, P.: Our Meeting with Gradual: A Good Strategy for the Iterated Prisoner's Dilemma. In: Artificial Life V, Proceedings of the 5th International Workshop on the Synthesis and Simulation of Living Systems, pp. 202–209. MIT Press, Nara, Japan (1997)
14. Rust, J., Palmer, R., Miller, J.H.: Behavior of Trading Automata in a Computerized Double Auction Market. In: Friedman, J., Rust, J. (eds.) The Double Auction Market: Institutions, Theories, and Evidence, pp. 155–198. Addison Wesley, Reading (1993)
15. Rosser, J.B.: On the Complexities of Complex Economic Dynamics. Journal of Economic Perspectives 13(4), 169–192 (1999)
16. Lindgren, K.: Evolutionary Dynamics in Game-Theoretic Models. In: Arthur, W.B., Durlauf, S.N., Lane, D.A. (eds.) The Economy as an Evolving Complex System II, pp. 337–367. Addison Wesley, Reading (1997)
17. Arthur, W.B., Holland, J., LeBaron, B., Palmer, R., Tayler, P.: Asset Pricing Under Endogenous Expectations in an Artificial Stock Market. In: Arthur, W.B., Durlauf, S.N., Lane, D.A. (eds.) The Economy as an Evolving Complex System II, pp. 15–44. Addison Wesley, Reading (1997)
18. Diekmann, A., Mitter, P.(eds.).: Paradoxical Effects of Social Behaviour: Essays in Honor of Anatol Rapoport. Physica-Verlag, Heidelberg (1986)
19. Friedman, J.W.: Oligopoly Theory. Cambridge University Press, England (1983)
20. Kauffman, S.: At Home in the Universe – The Search for Laws of Complexity. Penguin Books (1996)
21. Chiong, R.: Applying Genetic Algorithms to Economy Market using Iterated Prisoner's Dilemma. In: Proceedings of the 22nd Annual ACM Symposium on Applied Computing (SAC), pp. 733–737. ACM Press, New York (2007)
22. Sargent, T.: Bounded Rationality in Macroeconomics. Oxford University Press, Oxford (1993)

A Local Search Algorithm for a SAT Representation of Scheduling Problems

Marco Antonio Cruz-Chávez[1] and Rafael Rivera-López[2]

[1] CIICAp, Autonomous University of Morelos State
Av. Universidad 1001, Chamilpa, 62209, Cuernavaca, Morelos, México
mcruz@uaem.mx
[2] Technological Institute of Veracruz
Miguel Ángel de Quevedo 2779, Formando Hogar, 91860 Veracruz, Veracruz, México
rrivera@itver.edu.mx

Abstract. This paper presents the application of a local search algorithm for a logical representation of the Job Shop Scheduling Problem (JSSP). This logical representation represents the JSSP transformed as a satisfiability problem (SAT). The proposed algorithm uses a local search in a wide neighborhood. This algorithm, called Walk Wide Search - SAT, is a variant of the WalkSAT algorithm. This search is possible because the included tabu list prevents an excessive number of repetitions of movements during the search process. This paper describes the algorithm and compares results of Walk Wide Search - SAT to WalkSAT.

Keywords: Job shop, satisfiability, SAT formula, disjunctive graph, Reduced SAT Codification of JSSP.

1 Introduction

The complexity theory groups computational problems according to the inherent difficulty of solving them. This classification is used to classify decision problems [1], which require a yes/no answer in order to obtain their solution. The best-known problem in this classification is the satisfiability problem, which is classified as NP-complete. The objective of the satisfiability problem is to confirm or deny the existence of an assignment of truth-values for the literals of a logic formula (SAT formula) which make the formula true. A SAT formula is usually written in its conjunctive normal form (CNF) that has the following three features: (1) a conjunction C of clauses C_i, i.e. $C = C_1 \wedge C_2,\ldots, \wedge C_n$, (2) each clause C_i is a disjunction of literals $X_i \text{v} \ldots \text{v} X_k$, (3) each literal X_j is a Boolean variable (negated or not).

Some decision problems could be defined like discrete optimization problems. An instance of a discrete optimization problem is defined by the function $c : F \rightarrow R$, where F is the finite set of solutions that defines the problem instance, R is the set of discrete values that define each solution in F, and c is the objective function. In an instance of a discrete optimization problem, it is necessary to find the solution $f \in F$ for which $c(f) \leq c(y), \quad \forall y \in F$.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 697–709, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The Job Shop Scheduling Problem (JSSP) [2] is one of the most difficult problems within the NP-complete classification [1]. As a discrete optimization problem, it is an NP-hard practical problem [3] found in the area of manufacturing. For these two reasons, JSSP is a problem of great interest to the scientific community.

The JSSP is frequently treated as a discrete optimization problem; in this paper, the JSSP is treated as a decision problem based on its representation in the form of a satisfiability problem, in order to work with a proposed local search algorithm. By working in this way, the proposed local search algorithm looks for the satisfiability of a SAT formula that represents JSSP. The satisfiability of the SAT formula provides a feasible schedule of the JSSP. With this schedule, the proposed local search algorithm for the satisfiability problem could be applied in the search for solutions in a discrete optimization problem, as JSSP is generally treated.

Various methods have been proposed for manipulate the JSSP using several models. Two of the most commonly used models are disjunctive graphs [4] and integer programming [5]. The methods used with these models can be divided into two groups, local search methods and optimization methods.

Local search methods are iterative procedures that allow movement from one solution in F to another. These methods search for a local optimum in the solutions space of the problem through the use of a neighborhood function, which is defined as follows: Given a feasible point $f \in F$ in an instance of a problem, a neighborhood of f is defined as a set $N(f)$ of feasible points near f. The set $N(f)$ called the neighborhood f, indicates that each solution $f' \in N(f)$ can be reached directly from f in one step. In accordance with this the neighborhood is defined by the function $N: F \rightarrow 2^F$. Some of the most well-known local search methods are Hill-climbing, Steepest-descent, Iterative improvement, GSAT and WalkSAT.

Optimization methods try to find the best solution evaluating the objective function of the problem in order to find their maximum or minimum value. Generally, optimization methods use local search methods within their procedure. The methods based on Branch and Bound [6], Simulated Annealing [7], Genetic Algorithms (GA) [8] and Shifting Bottleneck [9], [10] are optimization methods, which try to determine the best solution according to definite criteria of the objective function. At the moment the line has been taken of working with hybrid algorithms, which combine in their procedure global and local searches. El-Mihoub et al. [11] and Shannon [12], present a revision of hybrid algorithms that combine the power of the GA and the local search. In this revision, it is affirmed that a GA could quickly locate the region which contains the global optimum, but it takes a relatively long time to locate the local optimum in the region of convergence. The combination of the GA and a local search method could accelerate the search in order to locate the global optimum. All of these works demonstrate that a GA combined with local search improves greatly their efficiency in a wide variety of problems. This type of GA combined with local search methods is known as the Memetic Algorithm [13].

Because optimization methods incorporate the use of local search methods into their procedures, local search methods that provide feasible solutions in an efficient manner can be very useful in providing a good starting point for the optimization methods [14].

Most of the methods used to find solutions to the satisfiability problem [15], [16], [17] are local search methods. This paper presents an algorithm of local search called

Walk Wide Search - SAT (WWS-SAT) for SAT problems, which can be applied to the JSSP encoded as satisfiability (SAT) using a codification format called reduced SAT [18]. The algorithm proceeds first to generate a propositional formula in conjunctive normal form called this SAT formula that represents the JSSP. Next, the algorithm looks for an assignment of truth-values for the variables in the SAT formula which satisfy the formula. If the satisfiability of the SAT formula is obtained, it implies that the result is a feasible schedule of the problem. In general, all the possible assignments of truth-values that satisfy a SAT formula form the set of feasible schedules of a JSSP instance. The paper presents experimental results of WWS-SAT which present it like a local search method very efficient and that one could apply as the base on the optimization methods in order to make them more efficient.

This paper is organized in five sections. Section one is the introduction. Section two describes the model of the disjunctive graph of JSSP. Section three presents the process of SAT codification for JSSP [18], which is explained briefly. Section four presents the proposed algorithm and an example of the assignment of truth-values to a SAT formula derived for an instance of JSSP. Section five presents the experimental results. Section six presents the conclusions and future work.

2 The Disjunctive Graph Model

The disjunctive graph model represents JSSP as a disjunctive graph G [4], [5], which is a 3-tuple where G = (N, E, A). N is the set of operations which includes two fictitious operations, the start and the end of the JSSP. These operations are represented in the nodes of G. E and A are two sets, edges and arcs (disjunctive and conjunctive) respectively. The set of edges (resource capacity constraints) is formed of subsets of these edges, each one of these subsets represents a machine M_i, E = $\{M_1, M_2, M_3, .., M_m\}$. The set of arcs (precedence constraints) is formed of subsets of these arcs, each one of these subsets represents a job, J_i, A = $\{J_1, J_2, J_3, .., J_n\}$.

Figure 1 represents the JSSP of two jobs and two machines (2x2). The problem has four operations, numbered 1 to 4, which are represented by the nodes of the graph. The start and the end operations are I and * respectively. The processing time of each operation is noted beside the corresponding node. Job 1 consists of operations 1 and 2, while job 2 consists of operations 3 and 4. P_1 and P_2 are the precedence constraints and they connect each pair of operations that belong to the same job. P1 means that in job 1, operation 2 cannot begin before operation 1 finishes. Likewise, P_2 prohibits the starting of job 4 until job 3 is finished. Machine M_1 executes the operations 1 and 4, while machine M_2 executes operations 2 and 3. R_1 and R_2 are the resource capacity constraints and they connect each pair of operations that are executed by the same machine. R_1 means that for M_1, operation 1 can be executed before 4 or 4 can be executed before 1, but the two operations cannot be simultaneously executed. Similarly, R_2 means that M_2 can carry out operations 2 and 3 in any order, but not simultaneously. In order to find a feasible schedule for this model, it is necessary to find the direction of each edge R, such that the formed schedule does not contain cycles.

3 Reduced SAT Codification

The reduced SAT codification of JSSP [18] is a SAT formula that contains a smaller number of clauses than other proposed SAT encodings for problems of scheduling [16], [17].

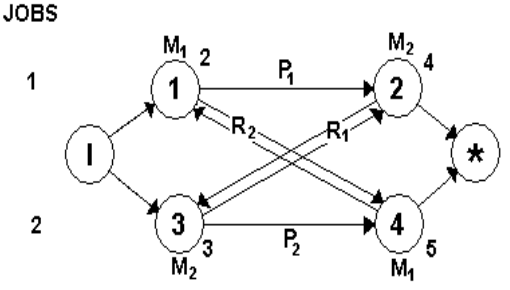


Fig. 1. JSSP of 2x2 represented by a disjunctive

Each clause in the reduced SAT codification is formed using only the restrictions of R and not involving the restrictions of P (Figure 1) due to the use of the latest starting time (LST) as the start time ***s*** of each operation [18]. The restrictions R, are defined as $s_i + p_i \leq s_j \vee s_j + p_j \leq s_i$ [5] for a pair of operations *i, j*, where *s* and *p* are the start time and processing time of the operation. The obtained clauses can be seen in Table 1; they are the result of mapping the resource capacity constraints as a set of variables that take values only of true or false. For each R, two clauses are obtained. In Table 1, it is shown that for each set of variables $sa_{k,t}$ where *k* represents any operation, the operation *k* begins at or after the time *t*. In addition, $sa_{j,t+pi}$ means that the operation *j* begins at or after the time $t+p_i$. $Pr_{i,j}$ means that the operation *i* precedes *j* if the truth-value that it possesses is true. If however the truth-value is false then *j* precedes *i*.

Table 1. Set of clauses that compose the SAT formula

Clauses of *R*	CNF	Interpretation
$sa_{i,t} \wedge pr_{i,j} \rightarrow sa_{j,t+pi}$	$\sim sa_{i,t} \vee \sim pr_{i,j} \vee sa_{j,t+pi}$	If *i* Starts at or after *t* and *j* follows *i* then *j* cannot start until *i* is finished.
$sa_{j,t} \wedge pr_{j,i} \rightarrow sa_{i,t+pj}$	$\sim sa_{j,t} \vee \sim pr_{j,i} \vee sa_{i,t+pj}$	If *j* Starts at or after *t* and *i* follows *j* then *i* cannot start until *j* is finished

In order to find a feasible schedule for JSSP, it is only necessary to find a truth assignment that satisfy the SAT formula which is formed by the set of clauses defined in Table 1. In order to be able to evaluate the set of variables sa_k and $sa_{j,t+pi}$, it is necessary to know the start time *t* of each operation. These times are calculated using the digraph that is obtained from the disjunctive graph that represents the JSSP. The digraph represents a sequence of execution of operations, which is generated with the truth-values of the set of variables $Pr_{i,j}$ assigned in random form. The time *t* (LST) of each operation *i* is equal to the critical path (longest path) [18] generated in the

digraph between the operation *i* and the operation initial *I*. When there is an assignment of truth-values that satisfy the SAT formula, a feasible schedule is obtained as a result.

4 The Local Search Algorithm

An algorithm of local search always requires the assignment of truth-values to the variables that compose the SAT formula, which is formed by the clauses in Table 1. In order to reduce the solution space where the search will be carried out, a set of control variables is designated. By designating a set of control variables, it is possible to determine the truth-values of the remaining variables. The variables used as control variables are the variables $pr_{i,j}$ (Table 1), because a finite number of sequences of pairs of operations in a problem exists. In contrast, by using the variables *sa* as the control variables, the situation becomes more complicated because the variables *sa* involve start times, of which there are an infinite number for each operation. By designating the variables $pr_{i,j}$ as control variables, and assigning truth-values to these variables, the formation of a sequence of the execution of operations in the problem results. With the defined sequence, a digraph of the disjunctive graph is generated which allows the LST for each operation to be calculated. Because the LST is calculated in this way, the first variable ($sa_{i,t}$) in each clause will always be true and its negation false ($\tilde{}sa_{i,t}$). The previous observation leaves the SAT formula as a formula 2SAT, that is, a formula where each clause contains only two variables ($\sim pr_{i,j} \vee sa_{j,t+pi}$). The LST and the number of clauses are obtained by simplifying the digraph. The simplification involves using only the Hamilton routes [19] that are generated for each machine. The calculation of the LST is then one of linear order [10], [19] and the number of clauses decreases because the number of edges R is decreased.

One could see that the local search algorithm proposed in this paper is an hybrid algorithm because it requires of an assignment of truth-values to the control variables ($pr_{i,j}$) and also LST must be calculated in order to be able to evaluate the remaining variables ($sa_{j,t+pi}$) of the SAT formula, this last is not common in algorithms for satisfiability. After obtaining the complete assignment of truth-values for the SAT formula, the satisfiability of the SAT formula is evaluated in a following step.

4.1 Example of Assignment of Truth-Values to the SAT Formula

Next is the presentation of an example of the assignment of truth-values to the variables of control in the SAT formula. This example is the 2x2 JSSP represented by the disjunctive graph found in Figure 1. The total number of clauses involved is determined by equation (1) and is equal to four clauses, where *nmaq* is the number of existing machines.

$$Clauses = 2\ nmaq\ (nmaq\text{-}1) \tag{1}$$

$$Variables\ of\ Control = nmaq\ (nmaq\text{-}1) \tag{2}$$

The number of variables of control (*pr*) is determined by equation (2) and is equal to two, one variable for each pair of operations. The possible assignments of

Table 2. Assignment of truth-values for the SAT formula of a JSSP of 2x2

Variables		Truth-values CASES			
C	D	1	2	3	4
$P_{1,4}$		*F*	*F*	*T*	*T*
	$P_{4,1}$	*T*	*T*	*F*	*F*
$P_{3,2}$		*F*	*T*	*F*	*T*
	$P_{2,3}$	*T*	*F*	*T*	*F*

truth-values are obtained using the relationship (*nmaq!*)*nmaq*. In this example, the possible number of assignments of truth-values is four, as shown in Table 2.

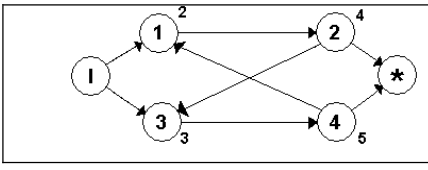
C are the variables of control and D are variables whose value depends on C. The resulting SAT formula is $\left(\neg \mathrm{Pr}_{1,4} \vee Sa_{4,t_1+p_1}\right) \wedge \left(\neg \mathrm{Pr}_{4,1} \vee Sa_{1,t_4+p_4}\right) \wedge \left(\neg \mathrm{Pr}_{3,2} \vee Sa_{2,t_3+p_3}\right) \wedge \left(\neg \mathrm{Pr}_{2,3} \vee Sa_{3,t_2+p_2}\right)$. As could be seen in this SAT formula, the number of Boolean variables $Sa_{k,tl+pl}$ is equal to the number of generated clauses. That means that the existence of a variable $Sa_{k,tl+pl}$ is observed for each existent operation in the problem, where *k= 1, 2, ... , nxm*, *n* is the number of jobs, *m* is the number of machines and *nxm* is the number of operations for a symmetrical problem of JSSP. The characteristic of generation of a small amount of Boolean variables in order to represent an instance of JSSP is characteristic of the reduced SAT codification [18], in this codification the variables $Sa_{k,tl+pl}$ are not generated in each unit of time $tl + pl$ as in other proposed SAT encodings for JSSP [16], [17]. Instead, variables $Sa_{k,tl+pl}$ only exist as defined by the units of time LST calculated in the operations of the problem. In order to evaluate a SAT formula only an LST for each operation is calculated.

When the four cases of assignment are proven in the SAT formula, it is found that in the first case, satisfiability of the formula is not obtained, but in the following three cases it is. A closer examination of case 1 and case 2 follows.

Case 1

The truth-values from Table 2 are assigned to the variables of control (C) and dependents (D). Then the LST of each operation can be calculated using the digraph generated by Figure 1 and Table 2. This digraph can be seen along with the LST of each operation in Figure 2. The start times of each operation grow to an infinite value because the operations in the digraph in Figure 2 belong to global cycles. Each operation is assigned a start time with a very high value, in this case 1000 is the designated start time. Substituting the times in each variable of the SAT formula results in the following: $\left(\neg P_{1,4} \vee Sa_{4,1002}\right) \wedge \left(\neg P_{4,1} \vee Sa_{1,1005}\right) \wedge \left(\neg P_{3,2} \vee Sa_{2,1003}\right) \wedge \left(\neg P_{2,3} \vee Sa_{3,1004}\right)$. Through the evaluation of each variable using Table 2 and Figure 2, it can be seen that the satisfiability of the formula is not obtained, that is (T $\vee$ F) $\wedge$ (F $\vee$ F) $\wedge$ (T $\vee$ F) $\wedge$ (F $\vee$ F).

Evaluation of Boolean variables $Sa_{k,tl+pl}$. In this case, all variable $Sa_{k,tl+pl}$ take the truth-value of false because the start time obtained by LST in each operation *k* grown to an infinite value.



LST
$t_1 = 1000$
$t_2 = 1000$
$t_3 = 1000$
$t_4 = 1000$

Fig. 2. Digraph and the assignment of LST in Case 1

Case 2

The truth-values from Table 2 are assigned as the control variables (C) and dependents (D). Then the LST of each operation can be calculated using the digraph generated by Figure 1. This can be seen along with the LST of each operation in Figure 3. By substituting the times in each variable of the SAT formula, the result is $(\neg P_{1,4} \vee Sa_{4,10}) \wedge (\neg P_{4,1} \vee Sa_{1,8}) \wedge (\neg P_{3,2} \vee Sa_{2,3}) \wedge (\neg P_{2,3} \vee Sa_{3,14})$. Through the evaluation of each variable using Table 2 and Figure 3, it can be seen that the satisfiability of the formula is obtained, that is (T $\vee$ F) $\wedge$ (F $\vee$ T) $\wedge$ (F $\vee$ T) $\wedge$ (T $\vee$ F).

Evaluation of Boolean variables $Sa_{k,tl+pl}$. In this case, the operation does not have a start time that grown to an infinite value (see Figure 3).

Clause 1. The Boolean variable $P_{1,4}$ takes the truth-value of false, this means that operation 1 does not precede operation 4 (see Figure 3), therefore, operation 4 should start before and end before or at the same time that operation 1 starts. This is not true because in clause 1 it can be observed that operation 4 starts in 10 and ends in 15 (processing time is added, see Figure 3) and operation 1 stars in 8 (see Figure 3) Therefore, the Boolean variable $Sa_{4,10}$ takes the truth-value of false.

Clause 2. The Boolean variable $P_{4,1}$ takes the truth-value of true, this means that operation 4 precedes the operation 1 (see Figure 3), therefore, operation 4 should start before and end before or at the same time that operation 1 starts. This is true because operation 4 starts in 3 and ends in 8 (processing time is added, see Figure 3) and in clause 2 it can be observed that operation 1 starts in 8, therefore the Boolean variable $Sa_{1,8}$ takes the truth-value of true.

Clause 3. The Boolean variable $P_{3,2}$ takes the truth-value of true, this means that operation 3 precedes operation 2 (see Figure 3), therefore, operation 3 should start before and end before or at the same time that operation 2 starts. This is true because operation 3 starts in 0 and ends in 3 (processing time is added, see Figure 3) and in clause 3 it can be observed that operation 2 starts in 3. Therefore, the Boolean variable $Sa_{2,3}$ takes the truth-value of true.

Clause 4. The Boolean variable $P_{2,3}$ takes the truth-value of false, this means that operation 2 does not precede the operation 3 (see Figure 3). Therefore, operation 3 should start before and end before or at the same time that operation 2 stars. This is not true because in clause 4 it can be observed that operation 3 starts in 14 and ends in 17 (processing time is added, see Figure 3) and operation 2 begins in 10 (see Figure 3). Therefore, the Boolean variable $Sa_{3,14}$ takes the truth-value of false.

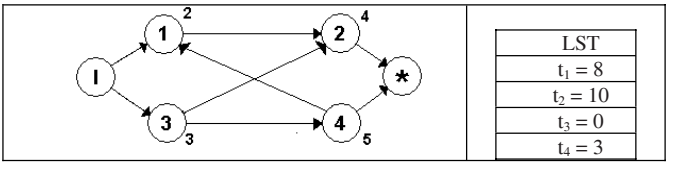


Fig. 3. Fig. 3 Digraph and the assignment of LST in Case 2

4.2 Walk Wide Search-SAT Algorithm

The local search algorithm, which is presented here and called Walk Wide Search-SAT (WWS-SAT), is based on the WalkSAT algorithm [20]. WWS-SAT is an algorithm that looks for the satisfiability of a SAT formula. This algorithm was developed for practical problems of Job Shop Scheduling. In contrast to theoretical problems [21], which are commonly used with the problem of satisfiability, the SAT formula generated from JSSP presents different characteristics. One such characteristic is that repeated variables do not exist. These differences cause changes in the behavior of the algorithms that are frequently used in the SAT area, which results in their being less efficient [16].

The WalkSAT algorithm begins with a randomly generated assignment of truth-values for the SAT formula. This assignment of truth-values is defined as Assignment-A. With a probability P, the search focuses on variables of clauses not satisfied. With a probability 1-P, the search focuses on variables of clauses taken randomly and chooses a change in the assignment of the truth-values of the variables. By changing the assignment of truth-values, the number of clauses that are not satisfied is diminished. The WalkSAT algorithm requires three parameters: (1) The set of clauses that generate the SAT formula to be evaluated, (2) the values of MAX-FLIPS that determine the number of exchanges that will be attempted, and (3) the value of MAX-TRIES that determine the number of times the search will restart before finishing. The algorithm presented here, WWS-SAT, can be described as a walk wide search because search in a wide neighborhood and carries out more than one movement in the neighborhood. In Table 3, it can be seen that WWS-SAT is a modification of Walk-SAT. With a probability P, WWS-SAT focuses the search on variables of clauses not satisfied. WWS-SAT also generates a list of all the clauses not satisfied and includes taboo movements as well. This list is defined as List-C. WWS-SAT eliminates from list the clauses in which there are repeated operations of the control variables pr_{ij}.

Table 3. WalkWideSearch-SAT algorithm

```
WWS-SAT(FormulaSAT){
 For i=1 to MAX-TRIES{
  A = a randomly generated truth assignment;
   For j = 1 to MAX-FLIPS{
    If A is a solution return;
     Else{
      C = Choosing the total of unsatisfied clauses
          where they don't repeat operations in pr;
          With probability P
             Flip variables pr_{i,j} in C;
          Otherwise (with probability 1-P)
             Flip a variable in an unsatisfied
               clause of C;
     }
   }
 }
 return failure;
}
```

For example, if there are three different clauses which are not satisfied, involving the variables $pr_{3,4}$, $pr_{4,5}$ and $pr_{5,6}$, with one variable *pr* in each clause, it can be observed that operation 4 repeats in the first and second variable and operation 5 in the second and third variable. If one of the clauses with variables that contain repeated operations is eliminated, the result could be $pr_{3,4}$ and $pr_{5,6.}$. This avoids falling into continuous cycles within a sequence of operations. This process enables the satisfiability of a SAT formula to be found in a shorter amount of time because the satisfiability of a SAT formula is a feasible schedule, which is a sequence of operations that does not contain any cycles. Once the List-C has been diminished by the elimination of clauses, the assignment of truth-values is changed for the control variables in remaining clauses of the List-C. It can then be seen whether assignment A is a solution or not. Alternately, with a probability 1-P, an unsatisfied clause can be taken randomly and the assignment of truth-values for the control variables can be changed. Then the Assignment-A can be tried out as a solution.

5 Experimental Results

In order to carry out the tests, a personal computer with a processor of 1.0 GHz and 1 GB in RAM was used. The operative system was Windows Millennium and the programming language was Visual C++ 6.0.

In order to prove the efficiency of the WWS-SAT algorithm, we carried out several comparative tests between it and the WalkSAT algorithm. The reported time in this paper is the time that takes the algorithm to satisfy the SAT formula, it does not include the time spent loading the program in memory or generating the SAT formula,

because the time to generate the SAT formula is very short because a reduced codification with few clauses is always generated [18]. The problems that were used in the tests can be found in the OR library [22]. They include: FT06 of 6x6, FT10 of 10x10, LA16 to LA20 of 10x10, ORB1 to ORB10 of 10x10, and LA36 of 15x15. For both algorithms, the probability used was P = 1. This was necessary because for a smaller probability, the results become much less efficient. The decrease in efficiency can be seen in Figure 4 where the graph of the problem FT06 using WalkSAT is presented. For each value of P, the number of flips in an average of 700 tests was obtained. It can be observed that as the value of P decreases, a higher number of flips are necessary in order to find the satisfiability of the SAT formula. As a consequence of the high number of flips, the time increases. The same behavior can be observed with WWS-SAT, so it was decided to use P= 1 for both algorithms when conducting the comparative tests. The results for WalkSAT and WWS-SAT used with problems of different sizes are shown in Tables 4 and 5 respectively. The presented results are averages of 2000 tests for FT06, 2000 for FT10, and 2000 for LA36. The percentage of achievements by each algorithm has a maximum of 100%. The number of flips, which are changes in the assignment of truth-values for a variable, is smaller in WWS-SAT than in WalkSAT. For FT06 the number of flips is 74% less, for FT10 it is 91% less, and for LA36 it is 6% less. The time to obtain the satisfiability of the SAT formula is also less for WWS-SAT. For FT06 the time is 81% less, for FT10 it is 45% less, and for LA36 it is 21% less. This indicates that in problems of varying sizes the efficiency of WWS-SAT is much better than that of WalkSAT. In addition, the efficiency of the generation of flips using WWS-SAT is better than that of WalkSAT.

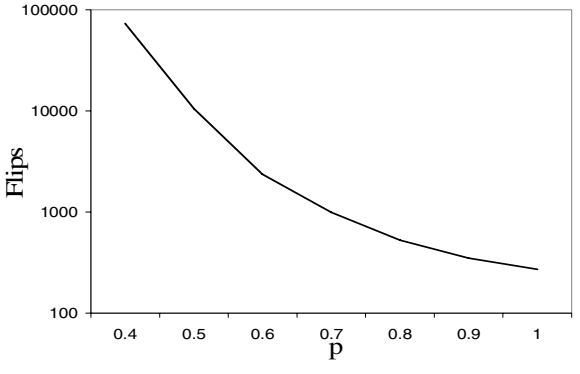


Fig. 4. Problem FT06. Flips vs. P. WalkSAT

Table 4. Experimental results in problems of several sizes for WalkSAT

WalkSAT		
JSSP	Success rate	Flips
FT06	100	00272
FT10	100	03853
LA36	100	47858

Table 5. Experimental results in problems of several sizes for WWS-SAT

WWS-SAT		
JSSP	Success rate	Flips
FT06	100	00072
FT10	100	00358
LA36	100	45066

Table 6 shows the results for the eighteen problems of 10x10 presented in the OR library. The results were obtained by running an average of 50 tests per problem. From Table 6, it can be noted that for the problems Orb3 and Orb8, it is difficult to satisfy the SAT formula with the WalkSAT algorithm because it takes an average of 0.017 and 0.01933 seconds respectively. That is not the case with WWS-SAT because the same problems only take 0.00124 and 0.00112 seconds to satisfy the SAT formula. It can also be observed that in most of the problems, WWS-SAT requires a smaller number of flips.

Table 6. Experimental results of the scheduling problems of 10x10, in OR Library

JSSP 10x10	WalkSAT		WWS-SAT	
	Flips	Time (s)	Flips	Time (s)
FT10	3968	0.00388	350	0.00176
Abz5	5495	0.00170	5071	0.00094
Abz6	5471	0.00163	5304	0.00096
La16	5113	0.00167	5117	0.00118
La17	5284	0.00160	5354	0.00116
La18	5652	0.00190	5212	0.00136
La19	5416	0.00157	5521	0.00126
La20	5486	0.00170	4741	0.00086
Orb1	4209	0.00476	366	0.00214
Orb2	5006	0.00167	5003	0.00136
Orb3	3259	0.01700	332	0.00124
Orb4	4521	0.00207	4305	0.00162
Orb5	4630	0.00280	4244	0.00190
Orb6	4321	0.00390	353	0.00207
Orb7	12631	0.00508	11819	0.00407
Orb8	3425	0.01933	341	0.00112
Orb9	4474	0.00200	4526	0.00136
Orb10	4551	0.00280	4547	0.00228

The exceptions are for LA16, LA19 and ORB9, which require fewer flips for WalkSAT. Even in these exceptions, the number of flips used by both WalkSAT and WWS-SAT is very similar but WWS-SAT wins in most of the problems, see FT10, Orb1, Orb3, Orb6 and Orb8 where the difference in flips is bigger. It should be noted that in all cases, WWS-SAT requires less time. By taking an average of all the problems listed in Table 6, it can be seen that WWS-SAT is more efficient because it generates 19% less flips and satisfies the SAT formula 63% more quickly than WalkSAT.

6 Conclusions

It has been demonstrated here that for tests carried out in practical problems like JSSP, WWS-SAT is superior to WalkSAT. For the SAT formula in JSSP, it is most convenient use a P = 1, due to the characteristics of the formula. This is because each pair of variables present in each clause does not repeat in any another clause. The use of a taboo list in WWS-SAT avoids returning to previous movements and avoids falling continually in sequences of operations that form cycles, consequently, allowing WWS-SAT to be applied to a bigger neighborhood. This larger neighborhood accelerates the acquisition of an assignment of truth-values that satisfy the SAT formula. For several problems of the same size, it is demonstrated that the use of WWS-SAT is uniform and does not change, as does WalkSAT. This constancy is achieved by the walk wide applied in WWS-SAT's local search. Instead of making a change in the assignment of a single variable, several changes are conducted in several variables that do not belong to the taboo list. These changes prevent stagnation in the local search, which happens in the search of WalkSAT because WalkSAT changes the assignment of one truth-value for one variable in one instant of time. The experimental results present to WWS-SAT like a local search method very efficient, because it find feasible solutions in a very simple form. This presents to WWS-SAT very attractive for their use in the optimization methods.

As seen when using the WWS-SAT algorithm (Section 4), the assignment of truth-values to the pr_{ij} variables is random. This allows for better distribution of the feasible schedules generated; they are located throughout the entire solution space of a JSSP. This characteristic could be of great interest for those who use Genetic Algorithms, because the better distribution of populations that are generated in the solution space could improve the search in a JSSP.

7 Future Work

WWS-SAT will be implemented with the algorithms of Simulated Annealing and GA in order to evaluate the efficacy of this local search method. In these optimization methods the Makespan as an objective function will be used in order to evaluate the efficacy of WWS-SAT. Efficacy will be evaluated as the ease with which solutions near the global optimum are found for the objective function in question.

References

1. Papadimitriou, C.H., Steiglitz, K.: Combinatorial optimization: algorithms and complexity, p. 496. Dover Publications, Mineola (1998)
2. Garey, M.R., Johnson, D.S., Sethi, R.: The complexity of Flow shop and Job shop Scheduling. Mathematics of Operations Research 1(2), 117–129 (1976)
3. Brucker, P.B.: Scheduling algorithms, 5th edn. Springer, Heidelberg (2007)
4. Balas, E.: Machine Sequencing via Disjunctive Graphs. An Implicit Enumeration Algorithm, Operations Research 17, 941–957 (1969)
5. Conway, R.W., Maxwell, W.L., Miller, L.W.: Theory of Scheduling. Addison Wesley, Massachusetts (1967)

6. Jalilvand, A., Khanmohammadi, S., Shabaninia, F.: Scheduling of sequence-dependant jobs on parallel multiprocessor systems using a branch and bound-based Petri net, Emerging Technologies. In: Proc of the IEEE, pp. 334–339 (September 17-18, 2005) ISBN: 0-7803-9247-7
7. Cruz-Chávez, M.A., Frausto-Solís, J., Zavala-Díaz, J.C., Sanvicente-Sánchez, H., Cruz-Rosales, M.H.: A Simulated Annealing Algorithm with Cooperative Processes for Scheduling Problems. LNCS. Springer, Heidelberg (to appear, 2006) ISSN: 0302-9743
8. Zalzala, P.J., Flemming.: Genetic algorithms in engineering systems, in A.M.S. Inst. of Electrical Engineers (1997)
9. Defu, Z., Tangqiu, Li., Shaozi, Li.: An improved shifting bottleneck algorithm for job shop scheduling problem. In: Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design. vol. 2, pp. 1112–1116, 24-26 (May 2005)
10. Schutten, M.J.: Practical job shop scheduling. In Annals of Operations Research 83, 161–177 (1988)
11. El-Mihoub, T.A., Hopgood, A.A., Nolle, L., Battersby, A.: Hybrid Genetic Algorithms: A Review, Engineering Letters (2006) ISSN: 1816-0948, 13:2, EL_13_2_11
12. Shannon Land, M.W.: Evolutionary Algorithms with Local Search for Combinatorial Optimization, Ph.D. Thesis, University of California, San Diego, p. 169 (1998)
13. Krasnogor, N., Smith, J.: A Memetic Algorithm With Self-adaptive Local Search: TSP as a case study. In: Whitley, Goldberg, Cantu-Paz, Spector, Parmee, Beyer (eds.) Proceedings of GECCO 2000, pp. 987–994. Morgan Kaufmann, San Francisco (2000)
14. Cruz-Chávez, M.A., Frausto-Solís, J.: Simulated Annealing with Restart to Job Shop Scheduling Problem Using Upper Bounds. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 860–865. Springer, Heidelberg (2004)
15. Ullman, J.D.: NP-complete scheduling problems. Journal of Computer System Sciences 10, 384–393 (1975)
16. Crawford, J.M., Baker, A.B.: Experimental Results on the Application of Satisfiability Algorithms to Scheduling Problems. In: Proc. of the 12th National Conf. on Artificial Intelligence, Austin, TX, pp. 1092–1098 (1994)
17. Memik, S.O., Fallah, F.: Accelerated SAT-based scheduling of control/data flow graphs Computer Design: VLSI in Computers and Processors, pp. 395–400, Proc IEEE (September 16-18, 2002)
18. Frausto-Solís, J., Cruz-Chávez, M.A.: A Reduced Codification for the Logical Representation of Job Shop Scheduling Problems. In: Laganà, A., Gavrilova, M., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) ICCSA 2004. LNCS, vol. 3046, pp. 553–562. Springer, Heidelberg (2004)
19. Garey, M.R., Johnson, D.S.: Computers and intractability: a guide to the theory of NP-completeness, p. 340. W.H. Freeman and Company, New York (1991)
20. Selman, B., Kautz, II, A.: Local search strategies for satisfiability testing. In: Procceding DIMACS Workshop on Maximum Clique, Graph Coloring and Satisfiability (1993)
21. Hoos, H.H., Stützle, T.: SATLIB: An Online Resource for Research on SAT. In: Gent, I.P., Maaren, H.v., Walsh, T. (eds.), pp. 283–292. IOS Press, Amsterdam (2000) 2006, SATLIB is available online at www.satlib.org
22. Beasley, J.E.: OR Library, Imperial College, Management School, Last Update (October 2005) (2007), http://people.brunel.ac.uk/~mastjjb/jeb/info.html

A Context-Aware Solution for Personalized En-route Information Through a P2P Agent-Based Architecture

José Santa, Andrés Muñoz, and Antonio F.G. Skarmeta

Department of Information and Communications Engineering
Computer Science Faculty
University of Murcia
Campus de Espinardo, 30071 Murcia, Spain
josesanta@dif.um.es, amunoz@um.es, skarmeta@dif.um.es

Abstract. Communication technologies and the management of context aware information are two key researching lines in the intelligent transportation domain. A suited network infrastructure which allows Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communications, on one hand, and the provision of location-based information adapted to the driver, on the other hand, comprise two important components in vehicle telematics field. Following this idea, our work includes a communication infrastructure based on Peer to Peer (P2P) networks which deals with V2V and V2I requirements. Reasoning agents have been integrated in the designed network, so the vehicle and the infrastructure's entities act as agents which adapt the information about the traffic area according to a driver profile. The paper describes also a prototype which implements both the vehicle edge, using an adapted vehicle, and the road edge, developing the network infrastructure and the knowledge management system. A performance study of the inference of contextual information has been added as well.

1 Introduction

Communications are essential in the current information society. Practically everyone can connect to Internet at work or at home, and even in mobility environments, using different kinds of portable devices. Cars are other places where the user stays during long periods too, so connectivity here is a new valuable field. However, apart from the direct usage of the Internet, some on-board services need specific communication primitives. The provision of context aware information to the driver is a key item at this point, where the synergy between the vehicular networks, the satellite based location technologies, and software agents that manage knowledge systems enable a new researching field.

Considering the connectivity requirements of services implemented on board vehicles, two types of communication necessities could be summarized: communications between the vehicle and the infrastructure, and communications among vehicles. In the first case, on-board services require a local connection with the

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 710–723, 2007.
© Springer-Verlag Berlin Heidelberg 2007

infrastructure located at the road side. This kind of connectivity is usually named as Vehicle to Infrastructure (V2I) communications. Some electronic fee collection systems use, for example, DSRC gantries to keep the path followed by a vehicle in order to calculate the associated cost. Here, the main technologies used are DSRC, infrared, and wireless LAN. The second group of such services are the vehicle to vehicle applications. Here, the most extended term is Vehicle to Vehicle (V2V) communications. Solutions for collision avoidance, and warning mechanism in general have been developed. For this type of services, the most extended technology is the ad-hoc networks applied to the vehicle field (VANET or *Vehicular Ad-hoc NETworks*). As can be read in the rest of the paper, the solution presented here proposes a communication architecture which deals with the these connectivity requirements. Hence, services with V2I and V2V necessities are considered in our proposal. Cellular networks are used, jointly with peer to peer (P2P) agent networks, to design a communication system based on coverage areas to reach such as architecture.

Using a suitable communication technology, vehicles could be connected to an information system which provides interesting notifications about the current status of a traffic zone. This kind of applications are common in mobile phones, where the current cell in the network indicates the position of the user, and therefore commercial advertisements about interesting places around can be sent to the mobile device. This model can be applied to the vehicle field. Basically, using a GPS device, the vehicle could be connected to a server which provides it with location based information about interesting places, for example. This approximation, however, does not take into account the user's preferences. Here is where the user's profile is important. If the system is able to adapt the information which is sent to the vehicle according with her profile, the driver would receive only the information she is interested in. In order to design a system with such features, not only a suitable knowledge representation model is necessary, but also an efficient inference technique is mandatory. The inference task must be carried out quickly, because now we are dealing with vehicles, that obviously move faster than a mobile phone carried by a person. As a result, the context information is updated more frequently.

Our work integrates a knowledge management model based on ontologies, using OWL (*Ontology Web Language*) modeling, to represent the information about interesting places in the traffic zones and the driver's profile. The ontology model allows the inference of interesting information about the zone according with the driver's profile, thanks to the formal logic language used in the description of interesting places or services of each zone and the driver's profiles. All the reasoning and data storage about interesting places and profiles are integrated in our communication architecture by means of the agents. Thus, as it is exposed along the paper, the data processing and data storage are distributed across the system.

The rest of the paper is organized as follows. In Section 2, the presented system is placed in the current researching world about vehicle communications and knowledge management. Section 3 explains the designed communication

architecture. Section 4 describes the integrated system for providing location aware information to drivers. The Section 5 serves to show the developed prototype. Finally, Section 6 contains the conclusions of our work.

2 Related Work

Regarding to vehicular communications, as it has been stated previously, the researching lines are centered in the usage of ad-hoc networks in the VANET field. The interest of this work lies in proving how cellular networks can be suitable not only for a direct Internet access, but also for enabling V2V and V2I communications [1]. In [2], a communication technique between the vehicle an the road infrastructure based on the DSRC technology is presented. This illustrates an example of a V2I communication method. The main drawback here is the cost of deployment at the road edge, which requires the installation of multiple DSRC devices, in contrast to our approach, based on cellular networks. In this line, [3] presents a handover system for wireless LAN networks in vehicles. Although the idea could be valid for urban or fixed railway zones, the deployment is limited by the cost. A work which captures a P2P approach through JXTA to notify events to the road side is shown in [4]. Our system use the JXTA technology as well in the vehicle field, however it also includes an architecture valid for V2V communications.

Delivering contextual information to the driver is the other main issue of the paper. The novelty of the system presented here is the inference of contextual information regarding to the driver's profile. Several approaches have been presented in other environments. Chou et al [10] develop a context-aware museum tour guide, in order to adjust its recommendations about the museum's items to the visitor's individual context. Similarly to us, they build the tour guide based on Semantic Web technologies such as OWL and web rule inference engines, which reduce the maintenance costs and the addition of new information. The rules are expressed in ROWL, a extension of OWL, whereas we use the Jena format for expressing our rules. A similar idea, but centered in touristic city tours, can be found in [11]. In this work, the mobile devices gather contextual information (location, preferences...) about the tourist and then it is sent to specific servers that process that information. These servers are Profile Management Server, for dealing with the tourist's preferences, and Location Server, that determines interesting touristic places on the tourist's location. The mobile devices use WiFi or GPRS as communication mechanism, whereas the information is represented using a own language based on XML. We advocate for the use of OWL as it is a W3C standard. Another work in this line is [12], changing the scenario to buildings and indoor spaces. In the vehicle field, [5] comprises a multi-purpose management system for contextual information. The system is exposed as valid for the vehicle domain and allows queries about the traffic conditions and route guidance, however it only presents an application for the urban environment, and does not adapt the retrieved information to the user preferences.

3 V2V and V2I Communications

3.1 A Multi-purpose Communication Infrastructure

It has been stated that cellular networks are the appropriate technology in order to use an unified communication mechanism. That is, cellular networks represent the communications capabilities which allow the implementation of any on-board service with connectivity requirements. However, an important issue in the service development arises when the amount of applications in the vehicle compartment grows. [7] works on this problem, and proposes a platform for the development of vehicle services based on a general purpose on board unit (OBU). A multi-platform system based on OSGi (Open Service Gateway initiative) is used for the deployment of services implemented as software. In the vehicle telematics field, a large number of services need communication capabilities. This is the reason why the work presented in [7] is extended with a middleware which allows the use of a high level communication interface. Using this mechanism, each new service does not require the implementation of its protocol starting from the TCP/IP basis. This common interface allows the message exchange in both V2V and V2I environments.

Using a P2P approach over cellular networks, the vehicle can receive and send contextual information about its current environment. Fig. 1 shows a general diagram of the high level communication architecture proposed, with the Group Server and Environment Server agents. The traffic zones are organized in coverage areas, each one of them using a different P2P communication group. Vehicles are able to move from one P2P group to another through a roaming process between coverage areas. This roaming is based on the location of the vehicle, provided by the GPS sensor. Information about areas is received from the Group Server using a TCP/IP link over GPRS/UMTS. The Group server stores the geometry of every service area. A local element called Environment Server manages special events inside the area. Event notifications are sent and received by service edges, located either at the vehicle or at the road side (Environment Servers). All messages emitted are encapsulated in P2P packages. Two different techniques of emission have been developed, so a P2P message can be broadcast in the area or sent to a specific vehicle.

Apart from the designed architecture, in Fig. 1 the three most representative scenarios are exposed from left to right. In the first one, a vehicle is passing from one area to another. The Group Server provides the P2P parameters to maintain the communication in all the active services. Because the Group Server sends the area geometry to the vehicle, the latter has to communicate with the central entity only when it is necessary, saving communication resources. In the second scenery, a safety service located in the vehicle notifies a repair event. This event is broadcast, so all vehicles located in the area receive the warning. Because the Environment Server also receives all the broadcast messages, it can process any event which requires a special treatment, such as a collision, for example. Messages from special services (repair, collision...) are forwarded by the Environment Server to the adjacent areas, in order to improve the warning

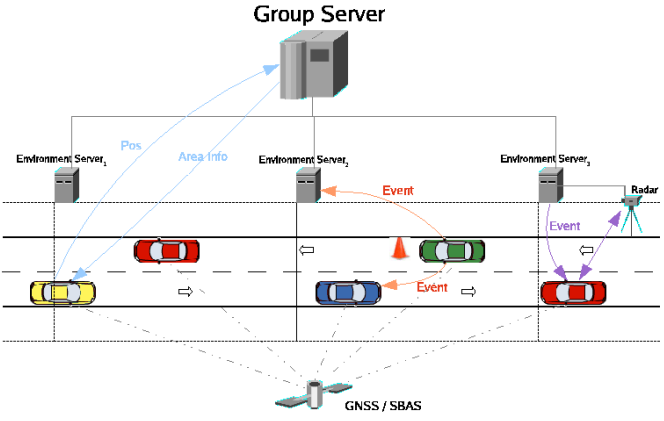


Fig. 1. High level communication architecture

mechanism. The last scenery in the diagram shows how the Environment Server can be connected to the rest of the road side infrastructure, which may be composed of speed radars, identification sensors, video cameras, etc. Thanks to this, local events can be notified to the vehicles in the area, and useful reports can be sent to the central entity.

3.2 Protocol Details

It is clear that the use of several protocols among the entities described in the previous section is necessary in order to perform all the processes that have been explained. To understand the operation of the whole system, the messages exchanged are divided into the three existing protocols detailed in the next lines.

Table 1 includes the messages exchanged between the vehicle (V) and the Group Server (GS). The communication is carried out by a message exchange over TCP, using the cellular network interface directly. The first process between these two entities is the management of the roaming. An initial *Area_Update_Query* message is used by the vehicle for registering itself against GS. The *Disconnect* message serves for ending the communication. The roaming process uses the *Area_Update_Query* and *Area_Update_Response* messages. The first one is used when a vehicle detects the end of its current coverage area. In that moment, it sends a new *Area_Update_Query* message with its new location. GS replies with an *Area_Update_Response*. This message contains the available services in the zone and the P2P parameters to allow subscribing to them. When the user wants to use a defined service, she sends an *Event_Subscription* with the services she is interested in. It is important to note that every service has a different P2P group in every area. This allows for distinction among coverage zones.

Regarding to the communication between the Environment Server (ES) and GS, Table 2 summarizes all the interchanged messages. The *ES_Registration*

Table 1. Vehicle-Group Server communication

Purpose	Messages	Sender
Communication Management	Area_Update_Query	V
	Disconnect	V
Roaming Process	Area_Update_Query	V
	Area_Update_Response	GS
Event Subscription	Event_Subscription	V

Table 2. Environment Server-Group Server communication

Purpose	Messages	Sender
Communication Management	ES_Registration	ES
	ES_Registration_OK	GS
	ES_Registration_ERROR	GS
	ES_Registration_Update	ES
	ES_Registration_Update_Response	GS
Message Propagation	Neighbour_Groups_Query	ES
	Neighbour_Groups_Response	GS

message is used by the ES to connect with GS. This packet contains the location of ES, so GS is able to search for the P2P parameters of all services available in the zone through the *ES_Registration_OK* message. If any error occurs, GS sends a *ES_Registration_ERROR*. This packet includes the reason of the error. Because the Environment Servers are static entities, which may be connected during long time periods, they need to update its configuration periodically. The *ES_Registration_Update* message performs this task. It only asks for the P2P parameters of the coverage zone. GS answers with this information using the *ES_Registration_Update_Response*. The second group of messages are used in the message propagation mechanism. Sometimes it is necessary to ensure that all vehicles near the problem receive critical events. The location of such problem may be close to the border of the coverage area. Consequently, it is necessary to propagate the associated message to the neighbouring areas. In order to perform this task, the Environment Server, which listens to all events, uses the *Neighbour_Groups_Query* message to ask for the neighbouring P2P groups for the service in question. GS replies with this information through the *Neighbour_Groups_Response*. Now, ES can forward the message to the neighbouring areas.

Finally, Table 3 includes the three messages used by the services to send messages. The first message we can see is *Vehicle_Event*. It contains a message sent from a vehicle service edge. All vehicles located in the area, and the Environment Server, receive these messages. The *Environment_Event* messages are sent by the Environment Servers. This is used, up to now, by the message propagation method described previously. The last packet, *Specific_Environment_Event*, allows for sending a message to a specific vehicle.

Table 3. Vehicle-Vehicle/Environment Server communication

Purpose	Messages	Sender
Message Passing	Vehicle_Event	V
	Environment_Event	ES
	Specific_Environment_Event	ES

4 Managing Context-Aware Information for Road Services

4.1 Information Management Through Ontology Modeling

The importance of context is receiving considerable attention in combination with mobile devices as these devices are particularly affected by environmental changes. A general motivation is that context-awareness can serve to compensate for the abstraction that is required in the first place to make systems accessible in changing environments and situations. Several advantages are acquired by modeling these kind of domains with ontological languages such as OWL. First, the information represented by the model can be easily and broadly exchanged among heterogeneous applications that understand this language. Another fundamental capability that is provided by adding semantics is the inference process. This process allows reasoning about the model, giving as a result the extraction of new knowledge that was implicit in the domain. This inference process is achieved by triggering some predefined set of rules about class hierarchy, types of properties, etc. Taking this idea further, we can define our own rule set that expresses our preferences about the behaviour of a semantically modeled application in each moment. Thus, this application will work according to the user's contextual situation.

Therefore, to deal with this type of scenario that involves context, location and preferences, we are going to make use of ontologies in modeling the user's profile and her context, the tracking location system explained in the previous section, and a set of special rules called *context rules*. Context rules model the user's preferences and they will have influence about the behaviour of an application depending on the user's context, for example the user's location, time of the day, relations with other users, etc. In this way, the same application may derive in different activities according to the context rules that are triggered when executing the application. These rules are formed by ontological elements, as it is explained in [13].

4.2 Ontology Modeling Adapted to the Generation of Context Aware Information in a Large Scale Environment

We have augmented semantically our P2P architecture by means of modeling the context aware information and the driver's profile using different ontologies that have been developed in order to satisfy the requirements of the system

commented along the paper. One of these requirements is its distributed composition. Consequently, the ontologies are spread among the different components of the architecture. Before explaining how this distribution has been designed, it is important to distinguish between the scheme and the instances of an ontology. Whereas there is only a scheme that represents the structure of the domain modeled, there could be various instances of that scheme for the concrete information captured. For example, the Profile ontology contains a scheme with the different preferences of a driver about restaurants, hotels, etc., and there is (at least) one instance of this scheme per each driver.

Going back to the distribution issue, the Profile ontology (scheme and instances) is stored in the GS. The possibility of that each car manages its own profile was studied, however due to performance results and the heterogeneous capabilities of the on-board units, we finally decided to move all profiles to the Group Server. In any case, we will keep on evaluating these two options and they will be present in our future work. By indicating the vehicle identifier, the GS returns the driver's profile, if it exists. On the other hand, the Service ontology (Fig. 2) is stored among the Environment Servers. This ontology models the context information of each area, i.e., the elements that could be of interest to the users. Each ES maintains the same schema, but the instances are specific for each zone controlled by them.

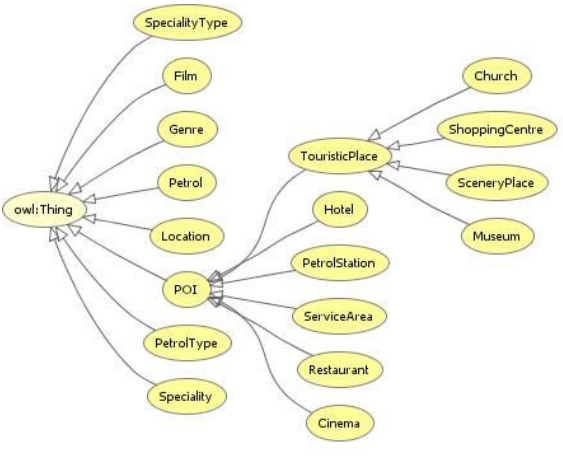


Fig. 2. Service Ontology in a hierarchical diagram

By combining the Profile ontology with the specific Server ontology of each ES, we obtain enough information to perform the inference process. Now, in order to incorporate the driver's preferences to the model, we make use of context rules. In Fig. 3 an example of these rules is shown. Concretely, this rule infers the restaurants which menu price is between a maximum *(le(?mp,?maxmp))* and minimum *(ge(?mp,?minmp))* value stated by the driver. If this condition is hold

by any restaurant, it is added to the list of driver's profile matches, specifying which condition is satisfied. Finally, the reasoning process is performed in the ES, as it will be explained later. One advantage of using context rules is that they can be used with any driver's profile or any specific instance of Service ontology. As a result, the same rules can be applied in any ES. Moreover, since the rules are defined in a external file to the source code, recompiling is avoided in case that we need to add new behaviours or change them.

```
|
# Check menuPrice
[menuPrice:
                (?profile rdf:type prf:Profile)
                (?profile prf:restaurantProfile ?rp)
                (?rp prf:maxMenuPrice ?maxmp)
                (?rp prf:minMenuPrice ?minmp)

                (?res rdf:type srv:Restaurant)
                (?res srv:menuPrice ?mp)

                le(?mp,?maxmp)
                ge(?mp,?minmp)
                ->
                (?profile prf:matches ?res)
                (?res srv:matchesWith 'menuPrice')
]
```

Fig. 3. A context rule for matching the driver's preferred menu price

In Fig. 4 a sequence diagram of the proposed system is shown. As it has been stated, the method by which the Environment Server detects the location of the vehicle is independent from the inferring and the knowledge model. In this approach, a RFID (*Radio Frequency IDentification*) system is used to detect a vehicle in certain locations. All the messages involved in the process of contextual information inferring are shown in the sequence diagram as well. First, the vehicle is detected by the RFID subsystem at a specified location. When the RFID reader detects the vehicle, it sends a *Reader_Notification* message. In this message, the RFID subsystem sends the vehicle identifier, so the Environment Server can use this information to search for the driver's profile. The message *User_Profile_Query* is addressed to do this task. The Group Server looks for the driver's profile in its data base and, if the profile does not exist, it replies with a *User_Profile_Not_Found*. In other case, if the profile is found, the Group Server replies with a *User_Profile_Response*. This message contains the driver's OWL profile. Now, the Environment Server processes the local information about the area, together with the driver's profile, to infer the interesting notifications in the zone. When this information is computed, a *Specific_Environment_Event* message is sent to the vehicle in question. The service edge at the vehicle is in charge of showing this information in an appropriate way.

5 Prototype and Tests

The system described in the previous sections has been implemented in a real platform. All technological details about the prototype are explained in this

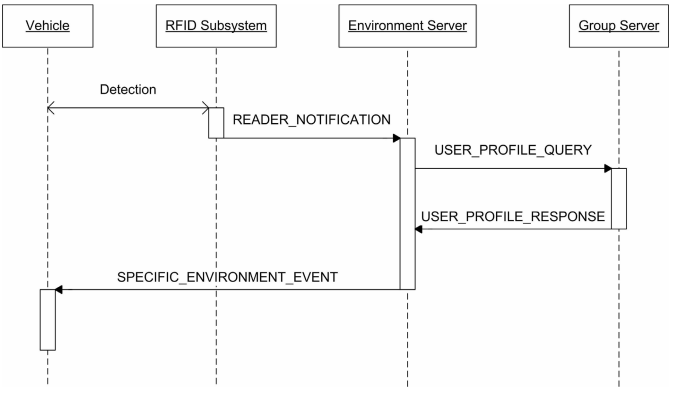


Fig. 4. High level communication architecture

section. The explanation is divided into the vehicle edge, the road side equipment used, and the software implementation at the road side and, mainly, at the vehicle edge.

5.1 Test Car and Road Side Equipment

The vehicle used for the system deployment is a vehicle widely sensorized at the University of Murcia. In [8] it is explained in detail all the hardware installed in the vehicle. The GPS sensor is utilized to know the position of the car and, therefore, it is essential for the roaming process among the coverage areas. The car in question is shown in Fig. 5. The on-board unit is a SBC (*Single Board Computer*) of VIA. The interface between the user and the OBU is carried out by a LCD panel installed on the dashboard. The operating system used in the OBU is a Linux Fedora Core 4, and a Java Virtual Machine 1.5 enables the computer to support all the multi-platform software implemented. In [7] it can be found more details about the software platform used in the OBU. Regarding to the communications, a cellular network PCMCIA transceiver has been used. The model is a Novatel Wireless Merlin U530, which allows GPRS and UMTS data connections.

At the road side, the tests have been carried out using a set of areas defined over the campus of the University of Murcia, as well as several kilometers of a close motorway. Eight areas has been used to probe the communications architecture. The Environment Servers have been implemented as software entities, running in a server located in one of our laboratories. The Group Server runs over another computer. All of them are Linux based systems, with a common Intel processor. A RFID system has been used to test the provision of contextual information. Using a tag located on the windscreen of the car, the reader can detect the presence of a vehicle. The tests in this case have been carried out in an open environment located at the campus of the University of Murcia.

Fig. 5. Prototype vehicle used in the development

5.2 On-Board Software Implementation

A new service has been implemented at the vehicle side to probe the feasibility of the communication architecture presented. JXTA (JuXTApose) [9] has been the P2P technology used. JXTA presents a group based communication system with the features of any other P2P protocol. It forms an underlying network which abstract the user from the internal details. Fig. 6 shows the Message Console service. Using the P2P middleware, an event based mechanism has been created to notify traffic incidences. Each "event" has been modeled like a service in the communications system. Eight types of events are available in the right box, so the user can subscribe to any of them. The most important service is named *On road information*. If any of the services is active, the corresponding events can be received or thrown. The events regarding to the on road information service cannot be thrown by a user, because it is managed by the road side infrastructure. All the received events are shown in the central part of the window.

In the example provided in Fig. 6, the driver has activated all the available services, but only three of them are available in the current area. An event regarding to the "On road" information service has been received. As can be noted, the driver's profile has been used in order to adapt some tourist information available in the current coverage area. This information has been printed in the window, showing the matches with the user's preferences. It is noticeable how *Guggenheim* matches with one hundred percent of the user likes, but the three other museums only match at a fifty percent.

5.3 Inference Performance

We are mainly concerned about how much time is employed at inferring and delivering interesting notifications to the car, since it could drive very fast and

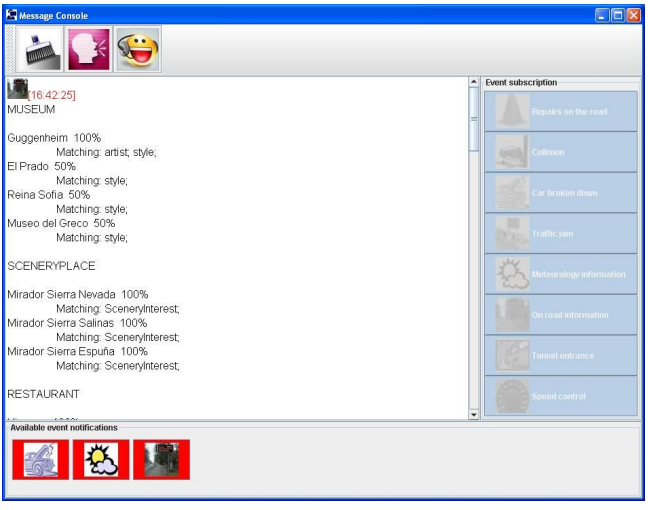


Fig. 6. An screenshot of the vehicle application

the notifications could arrive when leaving the corresponding zone. In order to evaluate the system performance, we have developed a model that contains enough information to deal with real situations and with all possible profiles. This model is composed of eleven and eighteen ontology concepts in the Profile and Service ontology, respectively. A Environment Server has been set up to simulate the zone where the car is driving along. This server contains one hundred and ten instances of different Service concepts, such as restaurants, cinemas, etc. Moreover, thirty-two context rules has been defined to match the driver's profiles with these concepts or services. Four different profiles have been proposed to measure the system responses. Each profile is centered on different types of services. Thus, Profile 1 is more keen on restaurants, Profile 2 on cinemas, Profile 3 on hotels, and Profile 4 is a mixture of these three. We have divided our performance test in three parts: First, the loading of the knowledge base (Service and Profile ontologies with the service instances and one driver's profile, together with the context rules) in the ES. Second, we measure the inference process, i.e., the application of the context rules over the knowledge base. Thirdly, the collecting matches task and the sending of information to the user is evaluated. The results are presented in Table 4. The ES employed is an Intel Pentium D 3Ghz with 1GB of RAM running under Linux, with the Jena [14] inference engine configured for the reasoning process.

The most expensive operation is the loading of the knowledge base, normally 2 or 3 seconds. However, since this process is performed once during the ES setting up, it does not affect to the normal operation of the system. More relevant are the results of the inference process and the collecting matches task, critical processes when a car is in the ES zone. As the profile is more complex, the inference process

Table 4. Evaluation results of the system performance

Profile	Load Knowledge Base	Inference Process	Collect Matches
Profile1	2484ms	250ms	62ms
Profile2	2625ms	281ms	63ms
Profile3	2890ms	313ms	47ms
Profile4	3410ms	381ms	105ms

takes longer, but it still stays inside of acceptable levels. However, the collecting matches task is unpredictable and variable, depending on the number of matches found. In principle, this is the faster operation, therefore the characteristics mentioned above do not represent a serious problem.

6 Conclusions

The paper exposes the work carried out in the synergy between the vehicular communications and knowledge management fields. A communication technique, based on cellular networks and the P2P protocol provided by JXTA, has been designed to offer the programmer a high level mechanism to use V2V and V2I communications in her on-board services. In addition, an integrated context-aware system to provide location-based information according to the driver's profile has been proposed. Semantic Web technologies as OWL have been selected to represent the information of the model, and an inferring process is followed to adapt information about the traffic zone according to the driver's profile. Finally, the designed prototype exposes the feasibility of he whole system.

In the future, several tests are planned to be made with the aim of testing the performance of the communication architecture in contrast to ad-hoc network solutions. Even though, this proposal is considered more general than ad-hoc ones, because it has a V2V message passing scheme based in P2P networks which also includes V2I capabilities. Another future line of work will be the evaluation of moving the driver's profile from the GS to the vehicle's OBU, in order to study the effects produced in the system's performance and reliability. This would offer the driver the possibility of changing the profile information dynamically.

Acknowledgments. The authors would like to thank the Spanish Ministerio de Educación y Ciencia and Fundación Séneca for sponsoring the research activities under the grants AP2005-1437 and 04832/BPS/06, in frames of the FPU and 2003-2006 PCTRM programs, respectively. Thanks as well to the financial support given by the European Spatial Agency (ESA) under the GIROADS 332599 project, and to the Spanish Ministerio de Fomento for its continuous support in vehicular researching.

References

1. Andreone, L., Provera, M.: Inter-vehicle communication and cooperative systems: local dynamic safety information distributed among the infrastructure and the vehicle as "virtual sensors" to enhance road safety, ITS Hannover 2005, Hannover (January 2005)
2. Hattori, G., Ono, C., Nishiyama, S., Horiuchi, H.: Implementation and Evaluation of Message Delegation Middleware for ITS Application. In: 2004 Symposium on Applications and the Internet-Workshops (SAINT 2004 Workshops), Tokyo (January 2004)
3. Okabe, T., Shizuno, T., Kitamura, T.: Wireless LAN Access Network System for Moving Vehicles. In: 10th IEEE Symposium on Computers and Communications (ISCC05), La Manga del Mar Menor (June 2005)
4. Baresi, L., Ghezzi, C., Miele, A., Miraz, M., Naggi, A., Pacifici, F.: Hibrid service-oriented architectures: a case-study in the automotive domain. In: 5th International Workshop on Software Engineering and Middleware (SEM'05), Lisbon (September 2005)
5. Chou, S., Hsieh, W., Gandon, F.L., Sadeh, N.M.: Semantic Web Technologies for Context-Aware Museum Tour Guide Applications. In: 19th international Conference on Advanced information Networking and Applications, Washington, DC (March 2005)
6. Vázquez, J.I., Lago, A.B., Larizgoitia, I., Uria, I., Bañales, R.: WALLIP: Una plataforma de servicios con percepción contextual basada en localización. Ubiquitous Computing and Ambient Intelligence, CEDI 2005, Granada (September 2005)
7. Falcó, J.M., Casas, R., Marco, Á., Falcó, J.: Using GUIA to "Wayfindig" in unkown builidings, Ubiquitous Computing and Ambient Intelligence, CEDI 200, Granada (September 2005)
8. Meier, R., Harrington, A., Cahill, V.: Towards Delivering Context-Aware Transportation User Services. In: 2006 IEEE Intelligent Transportation Systems Conference (IEEE ITSC 2006), Toronto (September 2006)
9. Nolte, T., Hansson, H., Bello, L.L.: Wireless Automotive Communications. In: Euromicro Conference on Real-Time Systems (ECTRS 05), Palma de Mayorca (July 2005)
10. Santa, J., Úbeda, B., Skarmeta, A.F.G.: A Multiplatform OSGi Based Architecture for Developing Road Vehicle Services. In: Consumer Communications & Networking Conference 2007 (CCNC 2007), Las Vegas (January 2007)
11. Muñoz, A., Vera, A., Botia, J.A., Skarmeta, A.F.G.: Defining basic behaviours in ambient intelligence environments by means of rule-based programming with visual tools. In: 1st Workshop of Artificial Intelligence Techniques for Ambient Intelligence, Riva de Garda (August 2006)
12. Skarmeta, A.G., Martínez, H., Zamora, M.A., Úbeda, B., Gómez, F.C., Tomás, L.M.: MIMICS: Exploiting Satellite Technology for an Autonomous Convoy. IEEE Intelligent Systems, N IV. V. 17, 85–89 (2002)
13. JXTA web site, `http://www.jxta.org`
14. Jena web site, `http://jena.sourceforge.net`

A Survey of Revenue Models for Current Generation Social Software's Systems

Kevin Chai, Vidyasagar Potdar, and Elizabeth Chang

Digital Ecosystems and Business Intelligence Institute, Curtin Business School, Curtin University of Technology, Perth, Australia
Kevin.Chai@student.curtin.edu.au,
{Vidyasagar.Potdar, Elizabeth.Chang}@cbs.curtin.edu.au

Abstract. In this paper we survey a number of different social software websites and analyze their revenue models. The main revenue models that we analyzed included - advertising, premium memberships, affiliate programs, donations and merchandize sale. The survey will categorize different social software based upon the revenue model being adopted. The main aim of this paper is to highlight the need for studying revenue sharing models which attempt to reward the users in the online community for participating in a particular social software website. A total of 7 revenue sharing websites would be discussed to show the importance of revenue sharing.

1 Introduction

Social software is software that allows people to socially interact with one another. The main idea behind the concept of social software, dates back to 1945, when the idea of people using computers to collaborate with one another was first proposed [1].The development of a computer assisted teaching system, called PLATO (Programmed Logic for Automated Teaching) in the 1970s gave birth to perhaps the first online community and it offered collaborative tools including discussion forums, e-mail, chat rooms, instant messaging, remote screen sharing and multiplayer games [54].

The term, "*social software*" was first introduced in 1987 by K. Eric Drexler in his paper, "*Hypertext Publishing and the Evolution of Knowledge*". In this paper, social software is referred to as "*filtering hyperlinks based on user voting and evaluation schemes*" in addition to facilitating group collaboration [14]. The term, social software, however did not gain popularity until late 2002, most likely attributed to Clay Shirky who organized the "*Social Software Summit*" in November 2002 [1]. From late 2002 or early 2003 we have seen a sudden interest in the adoption of social software. Social software mainly comprise of communication and interaction tools e.g. instant messaging programs, chat programs, discussion forums, blogs, wikis and collaborative real-time editors. These tools provide the building blocks to develop complex, robust and reliable social software services which include social networking, social bookmarking, video sharing, e-commerce as well as many other specialized services [40, 41].

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, pp. 724 – 738, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Although the concepts behind social software stem back from over 60 years ago, it is a term that is taking structure and it has been defined in many ways. [1, 15, 39]. One particular definition of social software states that it is "*software that allows individuals to collaborate, groups to self-organize and communities of individuals to evolve into an emergent structure*" [15]. In simpler words it could be treated as software that helps people to socialize online, form groups and allows for democratic approaches to the generation of online content [37]. Based on this definition, examples of social softwares would include websites such as *Wikipedia* [49] which allows the collective authorship of encyclopedic articles, *MySpace* [30] which provides people with the ability to create their own social networks and network with other people, *YouTube* [56] which lets people manage and share videos and, *eBay* [16] that empowers people to buy and sell goods or services from each other.

Clay Shirky provides a much broader definition and suggests that social software encompasses "*software that supports group interaction"* [17, 39]. Additional tools, not mentioned previously, based on this definition would include e-mail, instant messaging programs and chat programs, which allow for real-time communication (synchronous) between people and groups of people. Examples of these programs would include *Windows Live Messenger* and *Internet Relay Chat* (IRC). Shirky's definition intentionally includes offline community interaction as part of social software. Examples of offline social software would include *MeetUp* [29] and *nTag* [32]. *MeetUp* is a social software website that allows people to organise meetings and then meet in person for discussions. *nTag* provides interactive name badges that fosters social interaction amongst conference attendees through individual profiling, targeted networking by identifying people that share similar interests within the badge wearers proximity and the easy exchange of contact information via the badge.

Based on these definitions we can identify that one of the key characteristics of social software includes people that participate in an online (or offline) community, The question however arises, why are people motivated to use social software and why are they contributing? The answer to this question could be that people are motivated to engage in online interactions and contribute content with the goal of developing new social connections, maintaining existing ones and/or earn good reputation from others in a community [17].

Other than people, another important characteristic of social software is that it can experience *rapid growth* and popularity by allowing its users to create content e.g. *YouTube*, a video sharing portal is renowned as the fastest growing website in Internet history [45]. *YouTube beta* was launched in May 2005 and within a year it was streaming more than 3 million videos and receiving 8,000 user generated videos on a typical day [57]. This kind of active participation by end users in generating the content for a website is referred to as *user generated content* (UGC) and this can include items such as comments, articles, reviews, songs, pictures and videos etc. [44]. Another, popular and highly successful social software that has grown rapidly from UGC is *Wikipedia*, which is a free online encyclopedia that anyone can edit.

What we can understand from these examples is that the success of any social software or social service depends on the contributions made by the two key entities; the *infrastructure provider* and the *content provider*. The infrastructure provider provides the platform for a society to emerge and develop where as the communities in the society provide the content to add richness to the society. However, providing a

strong and robust infrastructure comes at a cost and the infrastructure provider may require sources of revenue to sustain the online community.

Currently social software relies on advertising, membership fees, affiliate programs, donations and selling merchandise, to recover most of their costs and to possibly generate a reasonable amount of profit. While the infrastructure provider generates revenue it would be reasonable to expect that some share of the revenue is passed to the active content providers for their effort. There are very few revenue sharing models currently deployed, and those that are deployed are not very realistic e.g. *VideoEgg's* [47] revenue sharing program is tailored for those content providers who received more than 50,000 unique visitors each month. This appears to be a very high expectation placed upon users; however it is not that the *VideoEgg* would not be making any revenue when it receives slightly less visitors than the prescribed limit. Although there are revenue sharing models in place, they are not really beneficial to the content providers.

The main aims of this paper is to *firstly*, study different revenue models adopted by social softwares and *secondly*, try to highlight the need for the development of realistic "*revenue sharing*" models that provide reasonable amounts of compensation to content providers who are actively involved in contributing to online communities.

The paper is organized as follows. Section 2 introduces a classification of business models for social software and examines a survey conducted on business models currently employed in social software. Section 3 explores issues in current social software business models. Section 4 concludes the paper with some future research directions.

2 Business Models for Social Software

In this section we review the business models adopted by several key social software. We review nine different categories of social software which include; Bookmarking, Collaboration, Community, E-commerce, Project Management, Text, Video, Wiki. Based on the review we have classified the business models for social software in two different categories which include "*free*" models and "*free and revenue*" based models. The *free model* apply to social software that do not receive revenue and the *free and revenue* based models derive revenue from the following approaches; targeted advertising, membership fees, affiliate programs, donations and selling merchandise. The following sections provide a detailed overview of the two business models.

2.1 Free

Free models refer to social softwares that do not generate any form of revenue. One of the greatest advantages in providing social software for free is that it attracts more people to trial the software compared to software that people have to pay for. If the software is well developed and provides value to users then existing users would likely assist the growth of the social software through word of mouth promotion to other potential users.

del.icio.us [11] is an example of a popular and free social software that was created in 2003 by Joshua Schachter as a hobby to allow an informal way to tag and share bookmarks between friends. *del.icio.us* was acquired by Yahoo! in December of 2005 and has so far decided to maintain this business model, but this does not mean that Yahoo! or other companies that may acquire del.icio.us in the future will not incorporate revenue models into *del.icio.us* [12].

CiteULike [9] is another example of a social software that allows academics to store, share and organize academics papers they are reading. Currently *CiteULike* is hosted by the University of Manchester and is offered as a free service. The creator of *CiteULike*, Richard Cameron states that if CiteULike becomes more popular and if the demand surpasses existing infrastructure capacity then options of funding may need to be incorporated. These options include; academic funding, user donations, licensing the server software and, targeted advertising. Cameron has been empathic to the *CiteULike* user community by requesting their feedback on the possibility of pursuing these options and to identify how users think these options will impact them [10]. It seems that user feedback is a critical component in deciding future directions for social softwares.

Glypho [20] is a free website that allows people to collaborate in writing novels. The *Glypho* novel writing process involves, firstly, for a person to post their ideas for a particular story. People from the *Glypho* community would then provide character and plot ideas. From these ideas, people interested in this story will write their own version of the first chapter. The community will then vote to select the best contribution for the first chapter. This process is then reiterated through subsequent chapters until the novel is completed.

Doostang [13] is a free social networking website that supports an online career community to connect people through both personal relationships and affiliations. It provides an ideal place for people to share job opportunities with one another, identify potential employees for employers and allows people to network with each other. *Doostang* also offers free premium membership accounts to users that have referred 20 new members to join the *Doostang* community.

We now discuss the free and revenue based models.

2.2 Free and Revenue

This business models is adopted by social software to generate revenue. We found five basic revenue models which include; targeted advertising, membership fees, affiliate programs, donations and selling merchandise. A brief explanation of each of these models is provided in Table 1.

It has been seen that revenue generating models are incorporated into social software to help cover costs such as hosting and staffing and to possibly accrue profit. These costs can be passed directly or indirectly on to social software users. Typically, social softwares that employ revenue models also offer products or services for free to help generate quicker growth and repeat usage of the software. A more detailed explanation of each revenue model with examples is explained now.

Table 1. Revenue models

Model	Description
Advertising	Advertisements are displayed to users of the software. The software providers can be paid based upon on amount of times the advertisement is displayed and/or the amount of times the advertisement has been clicked.
Membership	Membership and subscription fees can be imposed on users to access the software. Some social softwares may offer some functionality for free but charge a fee for access to premium functions.
Affiliate Programs	Affiliate partnerships can be established between the infrastructure provider and other parties. Referrals can then made to other party's products and/or services to users of the social software. The infrastructure provider may then receive a commission for the referrals or sales made from these referrals.
Donations	Donations can be received from the community of users or other parties.
Merchandise	Social software branded products can be developed and sold. Products may include items such as t-shirts, hats and coffee mugs.

2.2.1 Advertising

Advertising is a popular form of revenue generation that is used on many websites and social softwares. A possible reason behind the popularity of advertising as a revenue model is due to the fact that costs are not directly imposed on to users. Advertising, however can degrade users' perceived quality of the software and their overall experience if the user interface is cluttered with unsightly advertisements. To help address this issue, websites like *Ning* [31] and *WikiSpaces* [52] provide membership plans that allow users to remove the display of advertisements for their accounts. Other users however may benefit if the advertisements displayed are relevant to their interests.

Google AdSense [21] is one of the most renowned forms of advertising amongst websites as it displays customized advertisements based on the content of a particular webpage. For example, if the content for a particular webpage is about software development then, *Google AdSense* advertisements displayed on that webpage will be related to software development. Advertisers bid for keywords and pay Google each time one of their advertisement links are clicked (this is referred to as pay-per-click advertising) or when specific advertisements are displayed every 1,000 times (referred to as cost per impression). A percentage of this payment is passed on to the owners of the website. This helps create a beneficial partnership between both Google and infrastructure providers which could be considered as the key behind the success of *Google Adsense* program. Alternative advertising providers include *The Yahoo! Publisher Network* [55] which provides pay-per-click advertising and *BlogAds* [4] which adopts a more traditional form of online advertising, in that advertisers will pay for their advertisements to be displayed a certain amount of time on selected websites.

The survey conducted shows that Advertising is the most popular form of revenue generation amongst the surveyed social softwares In this paper, a total of 66 different social softwares were reviewed and 53 of those relied on some form of advertising. This amounts to a total of 80% of social software websites which used advertising as a form of revenue generation. A detailed list of these websites is provided in Appendix A. We outline a few examples here.

YouTube [56] is a website that allows users to store, share and manage their videos. *YouTube* has on average 100 million videos streamed a day, 65,000 new videos uploaded a day and roughly 13 million unique visitors per month [45]. *YouTube* was recently acquired by Google in 2007 and now uses *Google AdSense* advertisements as its form of revenue generation. Although it is expected that Google would acquire a sizeable amount of revenue from their advertisements, it should also be noted that there would be significant infrastructure costs in providing this service free to users. *Friendster* [18] and *LinkedIn* [26] are two other popular social networking websites that use advertisements as a form of revenue generation. *World66* [54] is a travel guide wiki that allows people to contribute and maintain travel related information and also uses advertising as a form of revenue generation.

2.2.2 Membership

The second most popular revenue model is offering fee based membership accounts. Membership fees can be imposed onto users for software usage or the use of exclusive software functions. For example, you may wish to create your own personal wiki at *PBwiki* [34] for free but you also have the option to have your account upgraded which will increase the storage space of your wiki, provide you with the ability to make your wiki private, implement encrypted Secure Socket Layer (SSL) access and other premium functions for a fee.

Some membership websites however do not provide free accounts but may provide free trial accounts to users in which users can trial the software for a number of days before making a decision to apply for a paid account. *ServerSideWiki* [38] and *SocialText* [42] are wiki providers that offer free trial accounts to users. This model is applicable to social software which is believed to be well designed, easy to use and effectively satisfies its target users' needs. In essence, it should also provide additional value when compared to free alternatives to be a successful revenue model to adopt. Some form of user support is generally provided by the infrastructure provider in adopting this revenue model.

The survey conducted shows membership to be the second most popular form of revenue generation. 16 social software websites imposed membership fees on users as a medium for revenue generation, which forms a total of 24% in the total number of social softwares surveyed. A few examples of these websites are outlined here.

CarbonMade [8] is a website that allows people to manage their portfolio or portfolio projects that include items such as photos, drawings and electronic art. *CarbonMade* offers a free account which allows users to manage 5 projects and upload 35 photos. Their professional plan is offered at $12 per month and allows users to manage up to 50 projects, 500 high resolution images and 10 high quality videos.

OurStory [33] is another website that allows people to capture, share and preserve stories about their lives based on a timeline format. *OurStory* offers both free and premium user accounts. Premium accounts provide additional functionality including the ability to create an unlimited amount of private circles in which stories are only visible to selected people, add up to six images and three videos for each story, gain access to using a rich text editor for publishing stories and access to technical support for a cost of $39.95 per year.

2.2.3 Affiliate Programs

Affiliate programs are the third most popular form of revenue generation adopted by the social softwares surveyed. These programs allow webmasters to generate revenue through commissions received from referring visitors to other people's products or services. These referrals are typically tracked through a uniquely generated Uniform Resource Locator (URL) address which contains the affiliate partner's unique ID. Affiliate programs are also advantageous to the referred company as it equips the company with an online sales force to make additional sales which may not have been made through their own marketing means. *Amazon* [2] is an excellent example of a company that offers affiliate partnerships to webmasters to help promote their products ranging from electronics, apparel, computers, books, DVDs and other goods. Google also offers affiliate programs for webmasters to promote new users to Google services. Referral commissions will then be paid out to the referrer once certain referral conditions have been met. For example, if a referred publisher signs up for *Google AdSense* [21] and earns $5 within the first 180 days of being signed up then the refrerrer will be credited with $5. Detailed information on Google's referral commissions can be found at [22].

The survey conducted shows that Affiliate Programs is the third most popular form of revenue generation. A total of 5 social websites used affiliate programs as a medium for revenue generation, which forms a total of 7.5% share in the social software surveyed. A few examples of these websites are outlined here.

Read It Swap It [35] is a website that allows book lovers to exchange books with one another. This website allows users to browse through catalogs of books and individual book webpages are presented with affiliate links to *Amazon* [2]. These links allow users to view *Amazon* reviews on these as well as to gain referral commissions from *Amazon* if users decide to purchase the book from the *Amazon* website.

Lib.rario.us [25] is a social media cataloging website that allows people to catalog and write reviews on books, DVDs, music and games. This website adopts the same strategy as Read It Swap It in linking product review pages back to the *Amazon* [2] website.

It appears that this revenue model is particularly applicable and popular in community review type social software websites that allow users to post reviews on products and services. A referral link can then be added for specific goods or services for each review.

2.2.4 Donations

Donation is another form of revenue generation model adopted a number of social software websites. Donations can be contributed by users or other parties to assist in

covering software costs and future software developments. This revenue model may not work effectively with the membership model as users will be unlikely to offer donations if they are already paying for software usage or the advertising model if advertising is used heavily throughout the software. Accepting user donations may also inflate users' expectations of the infrastructure provider to provide software support and/or continual development of the software.

The survey conducted highlights Donations to be the fourth most popular source of revenue generation. A total of 3 social software websites used donations as a form of revenue generation, which is a total of 4.5% share in the social software websites surveyed. A few examples of these websites are outlined here.

Wikipedia [49] is a website that allows for collaborative authorship of encyclopedic articles and has grown immensely in popularity. It currently hosts 5.3 million articles in approximately 250 languages [51]. It is interesting to note that *Wikipedia* is run entirely from user donations. In 2006, *Wikipedia* received approximately $1.5 million through donations and had outgoings of approximately $800,000 [50].

Butterfly [6] is a free service that allows people to save and categorize webpages by adding notes and tags. These notes and tags can then be shared with with other people. *Buterfly* is not supported financially by any organization and has requested donations from the community to cover hardware and administrative costs. In return for donations, donors will have their names added to a donation list that is displayed on the *Butterfly* website. For every €20 the font size of the donors name will be increased on the donations list webpage. Currently, *Butterfly* has not received any donations [7].

2.2.5 Merchandise

Finally, websites and social softwares can choose to develop and sell social software branded merchandise as a form of revenue generation. *Meetro* [27] is a location-aware Instant Messaging client and a real-time social network that that allows people to but also allows users to see and identify other users who are logged in nearby. As part of providing this service for free, *Meetro* also offers *Meetro* merchandise on their online store which includes items such as T-shirts, cups, hats, bags and buttons [28].

Bibli.ca [3] is a website that allows people to publish and share works including scripts, short stories, academic papers, poems and novels. The creator of *Bibli.ca* plans to publish a book compiled of the community works which is then presumed to be sold. Currently it is unknown whether profits generated from book sales would be distributed to its authors. The survey shows that selling merchandise is the least popular form of revenue generation amongst the social software population surveyed.

2.3 Survey of Different Revenue Models for Social Software Services

A survey was proposed in Section 1 to identify revenue models that are currently adopted by social softwares. A subset of the websites listed on a Web 2.0 website directory [48] was used to identify revenue generating social software websites and

Table 2. Survey of social software websites vs. adopted revenue models

Social Software Type	Advertising	Member-ship	Affiliate Programs	Donations	Merchandise
Bookmarking	8	0	2	1	0
Collaboration	6	1	1	0	0
Community	12	3	0	0	1
E-commerce	2	0	0	0	0
Project Management	1	4	0	0	0
Text	7	2	2	0	0
Video	12	1	0	1	0
Wiki	5	5	0	1	0
Total	**53**	**16**	**5**	**3**	**1**

their adopted revenue models. A total of 66 different websites was surveyed and from the results we can identify that advertising is, by a wide margin, the most popular form of revenue generation and, was adopted by 80% of the survey population. The membership revenue model followed with a 24% of websites. The survey results are displayed in Table 2. Please note that that social software websites have been categorized into social software types to assist in the display of the survey results. A detailed list of surveyed websites can be found in Appendix A.

3 Open Issues in Social Software Business Models

From the previous sections of this paper we identified from a survey of social software websites that the main forms of currently adopted revenue generation models are advertising, membership, affiliate programs, donations and merchandise sale.

It is also identified that a number of highly successful social softwares are web based and the contributions receive from their user community is the key behind their success. A few of these highly successful social software websites were discussed in earlier sections of this paper.

It is has been identified that social softwares can grow at viral speeds through the generation of content by regular repeat users and *word of mouth publicity* from existing satisfied users. User involvement in social software adoption is critical and the success of any social website depends largely on its users.

This raises a very important question - *Are users being rewarded handsomely for their contributions that they make in the online community or are the profits are only kept with the infrastructure providers?* From the survey that was conducted we found that only a small percentage of websites actually incorporated some kind of revenue sharing mechanism (10.6% or 7 out of 66 websites).

The term *revenue sharing* refers to passing on revenue, received from one of the previously mentioned revenue models, to the users that have helped generate that revenue. Some sort of contribution ranking scheme is required to identify how much revenue should be passed on to each contributing user. Implementing a revenue

sharing model can not only attract new users to social softwares but also encourage repeat usage from existing users and increases to the quality of their generated content so they can be ranked more favorably in terms of their contribution.

From the 66 websites that we surveyed we found that only 7 websites offered revenue sharing schemes. These websites include; Vizu, Ning, Broadband Sports, Pooxi, Videoegg, ZippyVideos and Revver. The revenue sharing model of these websites is briefly outlined in the following section. Additional details of these websites can be found in Table 3.

Table 3. List of Revenue Sharing Social Software Websites

Social Software	Social Software Type	Website
Vizu	Collaboration	http://www.vizu.com/
Ning	Community	http://www.ning.com/
Broadband Sports	Video	http://www.broadbandsports.com/
Pooxi	Video	http://www.pooxi.com/
Videoegg	Video	http://www.videoegg.com/
ZippyVideos	Video	http://www.zippyvideos.com/
Revver	Video	http://one.revver.com/

Broadband Sports [5] is a website that allows users to store, manage and share sports related videos. Currently this website generates revenue through Google advertisements and allows its users to earn rewards if they score as one of the top ten users for the week. User scores are calculated based upon the number of videos they upload and the cumulative score each video receives from other users. This clever incentive scheme helps encourage existing users to actively use the software and to produce quality content as this will help encourage them to attract more visitor views and higher video rating scores.

Ning [31] is a website that allows users to create their own online community. Ning adopts a revenue sharing model along with other revenue generating approaches. Firstly, Ning generates revenue from Google advertisements and monthly membership plans that allow users to secure their online community to selected users, use their own domain name, manage advertising within their online community and to increase their allocated storage space. Their revenue sharing model revolves around the advertisement management feature in their membership accounts as it allows users to display their own advertisements in which they can generate revenue from.

Revver [36] is another video sharing website that generates its revenue through advertisements. The advertising revenue received by Revver through video views is split 50% / 50% between Revver and the video creator. Sharers (users that promote other Revver user's videos) can also help advertise Revver videos and will receive 20% of the advertisement revenue. The remaining 80% is then split 50% / 50% between Revver and the video creator. Revver's revenue sharing model appears to be the most appealing revenue sharing model when compared to other revenue sharing websites surveyed in this paper.

The survey also found that the majority of revenue sharing websites were from the video sharing domain e.g. - Broadband Sports, Pooxi, Videoegg, ZippyVideos, Revver. A further investigation of the revenue sharing schemes would reveal important insights into the operations of such schemes.

This new trend of revenue sharing in social software raises another very important question – *what would the existing social software websites do if they wanted to change their revenue model or adopt revenue sharing?* This is a very important question, in particularly for social software websites that currently do not enforce any form of revenue generation e.g. – *CiteULike* [9], *Wikipedia* [49], *del.icio.us* [12]. If these websites begin to generate revenue then should a share of this revenue be shared with their users? Fundamentally, the success of social software is largely dependent on the contributions of their users within their online community.

If, these websites decided to share some of its revenue with its users then we are faced with interesting and challenging research question - *how do we determine user contributions for social softwares that do not currently incorporate a revenue sharing scheme?* This is a question that *YouTube* [56], could be attempting to answer as it has recently moved from a free business model into a revenue generating model (advertising) after being acquired by Google™. These are research questions that need to be addressed by the social software community.

4 Conclusion

In this paper we surveyed 66 different social software websites and analyzed their revenue models. They were selected from 8 different categories, which included; Bookmarking, Collaboration, Community, E-commerce, Project Management, Text, Video and Wiki. It was identified that the main revenue models were; advertising, premium memberships, affiliate programs, donations and merchandise sale. Some issues were then highlighted with the way revenue was distributed back to users which largely contribute to the success of social software websites. A total of 7 revenue sharing websites were discussed to show how some of these websites have rewarded their contributing users. These revenue models however are very simple and further controls may be required to detect and handle fraud. This initial study has opened several new research questions and directions; which would be the best combination of revenue model for certain types of social software, how would a well established free social software website handle the migration to adopting revenue generating models, how can we assess the contributions made by the users if a revenue sharing model is to be incorporated? The survey conducted in this paper was focused on social software listed on a Web 2.0 website directory [48], therefore there may be other websites listed in this directory but were not included because they do not satisfy the adopted definition of social software.

References

1. Allen, C.: Life With Alacrity: Tracing the Evolution of Social Software (2004) (Retrieved April 11, 2007) from http://www.lifewithalacrity.com/2004/10/tracing_the_evo.html
2. Amazon homepage, http://amazon.com/
3. Bibli.ca homepage, http://bibli.ca/

4. BlogAds homepage, http://blogads.com/
5. Broadband Sports homepage, http://broadbandsports.com/
6. Butterfly homepage, http://butterflyproject.nl/
7. Butterfly – Donate (n.d.). (Retrieved April 13, 2007), from http://butterflyproject.nl/about/donate.php
8. CarbonMade homepage, http://www.carbonmade.com/
9. CiteULike homepage, http://citeulike.org/
10. CiteULike: Frequenly Answered Questions (n.d.). (Retrieved April 11, 2007), from http://www.citeulike.org/faq/all.adp
11. Del.icio.us homepage, http://del.icio.us/
12. Del.licio.us: Team (n.d.). (Retrieved April 11, 2007), from http://del.icio.us/help/team
13. Doostang homepage, http://www.doostang.com/
14. Dexler, K.E.: Hyptertext Publishing and the Evolution of Knowledge. In: Proceedings of the 9th Conference on Hypertext and Hypermedia, pp. 87–120 (1987)
15. Dron, J.: Social Software and the Emergence of Control. In: Proceedings of the 6th International Conference on Advanced Learning Technologies, pp. 904–908 (2006)
16. eBay homepage, http://ebay.com/
17. Farnham, S., Kelly, S.U., Portnoy, W., Schwartz, J.L.K.: Wallop: Designing Social Software for Co-located Social Networks. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences, Track 4 – vol. 4 (2004)
18. Friendster homepage, http://friendster.com/
19. Green, D.T., Pearson, J.M.: Social Software and Cyber Networks: Ties That Bind or Weak Associations within the Political Organization? In: Proceedings of the 38th Annual Hawaii International Conference, pp. 117b–117b (2005)
20. Glypho homepage, http://www.glypho.com/
21. Google AdSense homepage, http://google.com/adsense/
22. Google AdSense Help Center (n.d.) (Retrieved April 11, 2007) from https://www.google.com/adsense/support/bin/static.py?page=common.html
23. History of social software – Wikipedia, the free encyclopedia 2007. (Retrieved April 11, 2007) from http://en.wikipedia.org/wiki/History_of_social_software
24. How much will I earn from referrals? Google AdSense Help Center (n.d.). (Retrieved April 16, 2007) from https://www.google.com/adsense/support/bin/answer.py?answer=25889
25. Lib.rario.us homepage, http://lib.rario.us/
26. LinkedIn homepage, http://linkedin.com/
27. Meetro homepage, http://meetro.com/
28. Meetro HQ Online Store 2007. (Retrieved April 13, 2007) from http://www.cafepress.com/meetro
29. MeetUp homepage, http://meetup.com/
30. MySpace homepage, http://myspace.com/
31. Ning homepage, http://ning.com/
32. nTag homepage, http://ntag.com/
33. OurStory homepage, http://ourstory.com/
34. PBWiki homepage, http://pbwiki.com/
35. Read It Swap It homepage, http://readitswapit.co.uk/
36. Revver Homepage, http://one.revver.com/

37. Rodriguez, M.A., Steinbock, D.J., Watkins, J.H., Gershenson, C., Bollen, J., Grey, V., deGraf, B.: Smartocracy: Social Networks for Collective Decision Making. In: Proceedings of the 40th Hawaii International Conference on System Sciences, pp. 90–90 (2007)
38. ServerSideWiki homepage - http://serversidewiki.com/
39. Shirky, C.: A Group Is Its Own Worst Enemy (Retrieved March 18, 2007) (2003), from http://www.shirky.com/writings/group_enemy.html
40. Shirky, C.: Ontology is Overrated: Categories, Links, and Tags. (Retrieved April 6, 2007) (2005), from http://shirky.com/writings/ontology_overrated.html
41. Social Software – Wikipedia, the free encyclopedia (Retrieved April 11, 2007) (2007), from http://en.wikipedia.org/wiki/History_of_social_software
42. SocialText homepage - http://socialtext.com/
43. The Australian Frequent Flyer – Membership Levels. (Retrieved April 11, 2007) (2007), from http://www.frequentflyer.com.au/cmps_index.php?page=membership
44. The monetisation (revenue model) of user generated content? (Retrieved April 3, 2007) (2006) from http://soapbox.technovated.com/2006/10/12/the-monetisation-revenue-model-of-user-generated-content/
45. Thomas, D., Buch, V.: YouTube Case Study: Widget marketing comes of age - StartUp Review Blog. (Retrieved April 2, 2007) (2007), from http://www.startup-review.com/blog/youtube-case-study-widget-marketing-comes-of-age.php
46. Berners-Lee, T. (Retrieved April 11, 2007) (2007), from http://www.w3.org/People/Berners-Lee/
47. VideoEgg homepage, http://videoegg.com/
48. Web 2.0 Trends: Web 2.0 Directory. (Retrieved April 2, 2007) (2007), from http://web2trends.blogspot.com/search/label/Web%202.0%20Directory/
49. Wikipedia homepage, http://wikipedia.org/
50. Wikipedia Foundation, Inc. Financial Statements. (Retrieved April 12, 2007) (2006) from http://upload.wikimedia.org/wikipedia/foundation/2/28/Wikimedia_2006_fs.pdf
51. Wikipedia: Size comparisions – Wikipedia 2007. (Retrieved April 12, 2007) from http://en.wikipedia.org/wiki/Wikipedia:Size_comparisons
52. WikiSpace homepage. http://wikispaces.com/
53. Wooley, D.R.: Plato: The Emergence of Online Community. (Retrieved April 11, 2007) (1994) from http://www.thinkofit.com/plato/dwplato.htm
54. World66 Travel Guide homepage http://world66.com/
55. Yahoo! Publisher Network homepage. http://publisher.yahoo.com/
56. YouTube homepage. http://youtube.com/
57. YouTube – Press Release December 2005 (2005) (Retrieved April 11, 2007), from http://youtube.com/press_room_entry?entry=OcN9xXYar1g

Appendix A: Survey of Social Software Websites vs. Adopted Revenue Models

Legend
R1 = Advertising
R2 = Membership
R3 = Affiliate Program
R4 = Donations
R5 = Merchandise
R6 = Revenue Sharing

Social Software	R1	R2	R3	R4	R5	R6	URL
Bookmarking							
9rules	X						http://9rules.com/
Blogmarks	X						http://www.blogmarks.net/
BuddyMarks	X						http://www.buddymarks.com/
ButterFly				X			http://www.butterflyproject.nl/
Digg	X						http://www.digg.com/
Lookmarks	X						http://www.lookmarks.com/
Ma.gnolia	X		X				http://ma.gnolia.com/
NewsCloud	X		X				http://www.newscloud.com/
Surf Tail	X						http://www.surftail.com/
Collaboration							
Asoboo			X				http://asoboo.com/
Favorville	X						http://www.favorville.com/
Hubpages	X						http://hubpages.com/
Lime	X						http://www.lime.com/
Rrove	X						http://www.rrove.com/
Standpoint	X						http://www.standpoint.com/
Vizu		X				X	http://www.vizu.com/
Wobblog	X						http://www.wobblog.com/
Community							
Wists	X						http://www.wists.com/
CarbonMade		X					http://www.carbonmade.com/
Homethinking	X	X					http://www.homethinking.com/
Stylehive	X						http://www.stylehive.com/
Consumating	X						http://www.consumating.com/
Friendster	X						http://www.friendster.com/
Ikarma	X						http://www.ikarma.com/
LinkedIn	X						http://www.linkedin.com/
Lovento	X						http://www.lovento.com/
Meetro	X				X		Http://www.meetro.com/
MySpace	X						http://www.myspace.com/
Meetup	X						http://www.meetup.com/
Ning	X	X				X	http://www.ning.com/
E-Commerce							
Ebay	X						http://www.ebay.com/
Esty	X						http://www.esty.com/
Project Managemnent							
Blue Dot Buzz	X						http://bluedot.us/
Basecamp		X					http://www.basecamphq.com/
CentralDesktop		X					http://www.centraldesktop.com/
Near-Time		X					http://www.near-time.net/
Project Spaces		X					http://www.projectspaces.com/
Text							
IloveTravelStories	X						http://ilovetravelstories.com/
LiveJournal	X	X					http://www.livejournal.com/
Reader2	X						http://reader2.com/
Read It Swap It			X				http://www.readitswapit.co.uk/
OurStory	X	X	X				http://www.ourstory.com/

Social Software	R1	R2	R3	R4	R5	R6	URL
World66	X						http://www.world66.com/
30daytags	X						http://www.30daytags.com/
TagCloud	X						http://www.tagcloud.com/
Video							
Blinkx	X						http://www.blinkx.com/
Broadband Sports	X					X	http://www.broadbandsports.com/
ClipShack	X						http://www.clipshack.com/
Dailymotion	X						http://www.dailymotion.com/
GetDemocracy		X		X			http://www.getdemocracy.com/
Openvlog	X						http://www.openvlog.com/
Pooxi	X					X	http://www.pooxi.com/
Video Bomb	X						http://www.videobomb.com/
Videoegg	X					X	http://www.videoegg.com/
Bolt	X						http://www.bolt.com/
YouTube	X						http://www.youtube.com/
ZippyVideos	X					X	http://www.zippyvideos.com/
Revver	X					X	http://one.revver.com/
Wiki							
Epic Trip	X						http://www.epictrip.com/
Schtuff	X						http://www.schtuff.com/
PBWiki	X	X					http://pbwiki.com/
Social Text		X					http://www.socialtext.com/
Wikispaces	X	X					http://www.wikispaces.com/
StikiPad	X	X					http://www.stikipad.com/
ServerSideWiki		X					http://www.serversidewiki.com/
Wikipedia				X			http://wikipedia.org/
Total	**53**	**16**	**5**	**3**	**1**	**7**	

Context-Driven Requirements Analysis

Jongmyung Choi

Dept. of Computer Engineering, Mokpo National University,
61 Cheonggye, Muan-gun, Jeonnam 534-729, Korea
jmchoi@mokpo.ac.kr

Abstract. Context-awareness is an important feature of ubiquitous computing systems. Compared to traditional systems, context-aware systems have different features such as heterogeneity, high complexity, and artificial intelligence, so that the need for its own development methodology increases. In this paper, we define context and context-aware service as both concrete and evaluative forms, and introduce a requirement analysis process, context-aware use case diagram, context-switch diagram, and dynamic service model based on those definitions. We expect that our work will help analysts and designers to understand, analyze, and design context-aware systems more efficiently.

Keywords. Context, Requirement Analysis, Context-Aware Use Case, Context-switch Diagram.

1 Introduction

One of the most important issues of ubiquitous computing systems is about the context-awareness, because it enables these systems to provide users with convenient services without users' intervention. User identity, location, and time are good examples of context, and systems are able to choose their services based on their current context. For example, a city tour guide system [1][7] may provide travelers with tour information based on their location.

Context-aware systems are complex, and they require various technologies including ergonomics, human-computer interface, artificial intelligence, sensor and sensor network, software engineering, and even sociology [2][3]. These widespread technologies make it more difficult for stakeholders to communicate about, analyze, design, and implement the systems. For example, customers and developers have communication problems in the requirement phase, and they face these questions: What context does the system under development need? What sensor should be used for perceiving context? What context triggers what services? And what is the difference in the services according to context? These communication problems and the difficulties in understanding context-aware systems are caused by the absence of requirement analysis based on precise definitions of context and context-aware service.

In this paper, we define context and context-aware service types as evaluative forms, and introduce a context-driven requirement analysis method. First, we define context as a precise and evaluative form because it is the key element to understand context-aware services and systems. Second, context-aware service type is important

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 739–748, 2007.
© Springer-Verlag Berlin Heidelberg 2007

in understanding context-aware systems. The service type is the category of services that are provided to users based on current context, and we classify those services into five types: branch, trigger, resource scanning, follow-me, and context recording. Context and context-aware service types are closely related to system usage patterns, so that they are very important to users, customers, analysts, and developers. Finally, we introduce a context-driven requirement analysis method. Because context-aware systems have complex and multiple aspects to consider, we need a new approach to capture the requirements of these systems. We introduce requirement analysis process, context-aware use case diagram, context-switch diagram, and dynamic service model to capture the various aspects of context related complexity.

The rest of the paper is organized as follows. In Section 2, we discuss the related works. Then we define context related concepts, and we classify context-aware services into five types in Section 3. After that, we introduce a requirement analysis method for context-aware systems in Section 4. Finally, we reveal the conclusions of our work in Section 5.

2 Related Works

Our work has relatively close relationship with two categories of existing studies: context taxonomy and requirement engineering for context-aware systems. First, there have been many researches about the concept and features of context [2][4][5] [8]. Schilit [8] defines context as location, person or object identifier, and the change of object. Dey [4] defines context as "any information that can be used to characterize the situation of an entity." In addition to these two definitions, there are many other definitions for context as surveyed in Dey's work [4], but those definitions are categorized into two types. One is defined as the list of example information, and the other is defined as rather vague and not evaluative terms. These definitions are not suitable for software engineering process. However, our definition is not only abstract but also concrete enough to be evaluated.

Second, there have been a couple of works on the requirement analysis for context-aware systems. Hannes [9] noticed the importance of context in use case creation, and so introduced a process to capture context situations. The main goal of the process is bridging the gap between critical context situations and the user goal by identifying the situations. Lyubov [10] introduced requirement engineering for pervasive services, and proposed a method for obtaining requirements for context-aware systems under development. In comparison to these works, we introduce a more concrete and comprehensive analysis process and analysis tools.

Aside from these two categories, there have been some researches on other software engineering subjects for context-aware systems. Dennis [11] introduced a context-driven refinement mechanism for component-based development and a context-aware modeling language similar to UML class diagram. Antonia [12] introduced the software architecture for context-aware systems in formal approach. These works do not have a direct correlation with our work, but in the perspective of software engineering, there is some relationship with those works.

3 Context-Aware Systems

3.1 Context

Although the definitions of context [4][6][8] vary according to researchers, they have common concepts. First, context is closely related to the interaction between human and computer. That is, a context-aware system senses an external or internal change, and it responds to the change interactively. Second, context has information, and it may consist of several attributes. For example, a typical context such as user location, time, and temperature is information, and its value determines which services must be provided.

According to the second feature specified above, context may consist of several pieces of information or attribute. In that case, we call the most basic component of context by the name of the context attribute, and it is the attribute value sensed from the external world via a sensor or the internal state value stored in a variable. Therefore context attribute is defined as Def. 1.

Def. 1. Context Attribute
Context attribute, a_i, is an atomic value sensed via a single sensor or a simple value stored in a variable. □

Context is an abstraction that represents the composition of external and internal conditions surrounding a user and a system. Furthermore as mentioned above, context is closely related to information or attribute, and we can define the abstraction, i.e. context, as a tuple of context attributes. Thus using context attributes, we can define context as follows.

Def. 2. Context
Context, c, is an abstraction of the external and internal conditions surrounding a user and a system, and it may invoke or determine context-aware services. Context is implemented as a tuple of the finite number of context attributes: $c = <a_0, a_1, \ldots>$. □

Some contexts have a common concept and may be represented by a high level abstraction. We call it context type. It is a kind of data type of context.

Def. 3. Context Type
Context type, t, is the high level abstraction that represents a common concept of contexts. It may be considered as the data type for context. □

Most context-aware systems have several situations in their lifecycle, and in each situation they provide services based on the current context. Therefore a context-aware system has several contexts, and we call the set of contexts in a system the context set.

Def. 4. Context Set
Context set is a set of contexts used in a context-aware system. □

Context-aware systems are intelligent. Furthermore some advanced systems may infer new context attributes or contexts from the sensed raw data. For example, a system may sense user identification, and infer his/her age, preference, or today's schedule.

We define this inference or interpretation function of context-aware systems as context inference function.

Def. 5. Context Inference Function
Context inference function is a function, $\delta(c)$, which can derive a context set from a known context, c. □

A context-aware system's context set may have many contexts as its elements. Furthermore, we can extend a context set by applying context inference functions dynamically. In contrast, we can think of the minimal set of context set of a system, and we call it the normal context set.

Def. 6. Normal Context Set
Normal context set, C_{base}, is the minimal context set. Context set, C, is the result of applying the context inference function to the normal context set, i.e., $C = C_{base} \cup \delta(c \in C_{base})$. □

In general, managing and processing context-attribute and context require several tasks: transforming the raw data into context-attribute, integrating several context-attributes into a context, interpretation of data, and inferring new information. These tasks are common in most context-aware applications, so that they have been developed as middleware or toolkit libraries. We call the middleware the context handler or context middleware.

Def. 7. Context Handler
Context handler is a software module that transfers sensed raw data to the application layer, integrates them into context, and infers new information from context attributes. □

3.2 Context-Aware Services

We call a system context-aware "if it uses context to provide relevant information and/or services to the user [4]." In the perspective of the interaction between human and computer as specified in section 3.1, a context-aware service is the system's response to the change of context. As context-aware services influence the system's usage scenarios and architecture, we need to categorize them to get their features. Therefore, we classify context-aware services into five groups according to context's role in the systems as follows.

- **B**ranch: context determines a service among several ones, its contents, or its output format.
- **T**rigger: change to context triggers the execution of a service.
- Resource **S**canning: the system scans the external resources and provides services that are combined with external resources. Context plays the role of medium of cooperation.
- **F**ollow-me: the system keeps user's context, follows him/her, and provides services with the help of surrounding context. Context plays the role of the medium that connects the past and the present.
- Context **R**ecording: the system records and saves the current context for later retrieval.

The service types listed above are not exclusive, so that a service can be included in one or more groups. For example, a context-aware service named "A" is triggered on the change of context, and the contents of the service are determined by context. Then the service is included in both Branch type and Trigger type.

In the perspective of context's interaction feature, a context-aware service is characterized with context and context-aware service types. Therefore, we define a context-aware service as follows.

Def. 8. Context-Aware Service
A context-aware service, *f*, is the service which is influenced by context in the perspective of service contents, service representation style, service triggering, and other service related features. The service is matched to one or more context service types. Thus it is defined as follows: $f(c_i) \rightarrow q_k$, where c_i is context and q_k is an instance of context service types. □

A context-aware system provides multiple context-aware services according to the context set of the system. The set of context-aware services is also an important and interesting point in developing context-aware systems. We define the set as follows.

Def. 9. Context-Aware Service Set
The context-aware service set of a system is the set of context-aware services that are supported by the system. □

4 Context-Driven Requirement Analyses

4.1 Requirement Analysis Process

It is very difficult and complex to capture the right requirements for context-aware systems because business logic is intermingled with context logic. Context logic has a close relationship with implicit interaction with users and the external environment, and it also has relationship with business logic because it triggers or determines the business services. Fig. 1 shows the relation among business logic, context logic, and user interface.

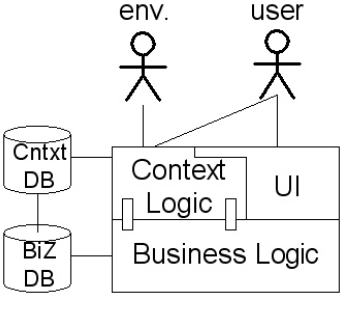


Fig. 1. Layer of Context-Aware Systems

Separation of concerns is usually used to reduce complexity, so we separate context logic from business logic in the requirement analysis phase. By separating concerns, we capture the requirements through two big steps: one for business logic, and the other for context logic. To get the right requirements, we must do customer interviews, workshops, and other activities as needed. The following process shows the steps to capture the requirements. Capturing and creating business logic use cases are not our main concern and they are discussed well in other works, so that they are noted briefly in step 1.

1. Create business logic use cases for the systems
2. Identify context-aware services in use cases
3. Identify context from the context-aware services
4. Identify the context-attributes that constitute context
5. Identify the subsystem or device that perceives the context attributes
6. Repeat steps 2-5 until no more context-aware services are left
7. Determine the system architecture

Identifying context-aware services is based on the context-aware service scenario and context-aware service types defined in section 3.2 which may be used as a guideline to identify the services.

1. Check the business usage scenario to determine whether it can be categorized into one of the context-aware service types
2. Add the service to the context-aware service set if it is one of the context-aware service types
3. Identify context for the context-aware service
4. Add the context to the context set for the system
5. Repeat steps 1-4 until no more context-aware services are identified

4.2 Context-Aware Use Case Diagram

Use case and use case diagram are widely used to elicit functional requirements from customers, and they are also used as communication tools among stakeholders. Not only business logic but also context and context-aware services have a close relationship with usage scenario, so they also must be described in the use case and use case diagram to fully understand the system under development. Furthermore, because they may have influence on the system architecture, it is important to describe them in detail as early in the process as possible. Because of these reasons, context and context-aware service types must be described explicitly in use case.

A traditional use case diagram does not support context-aware features, so we extend it by adding icons. Context-aware use case diagrams will help stakeholders to understand the system's context related requirements. Fig. 2 shows an example of a context-aware use case diagram. The <<sensor>> and <<state>> actors mean the source of context attribute. The <<sensor>> is the external source of context attribute, and the <<state>> is the internal source of context attribute. In the figure, the context-aware use case is labeled "B," and it means the context-aware service type of the use case is the Branch type. We can mark other symbols according to the service types: T(Trigger), S(Resource Scanning), F(Follow-me), and R(Context Recording).

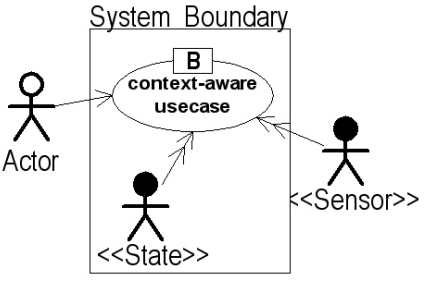


Fig. 2. Context-Aware Use Case Diagram

The context-aware use case diagram is very useful in understanding a context-aware system. Let's assume that we are developing a context-aware mobile phone introduced by Daniel [13]. If the phone does not support the context-aware features, its use case diagram will be rather simple as partly illustrated as Fig. 3. The phone notifies the phone call to callee, and he/she talks with the caller.

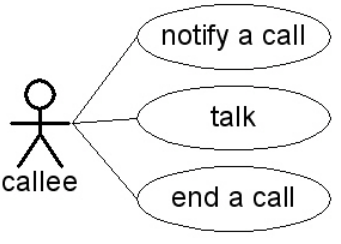


Fig. 3. Use Case Diagram for Mobile Phone

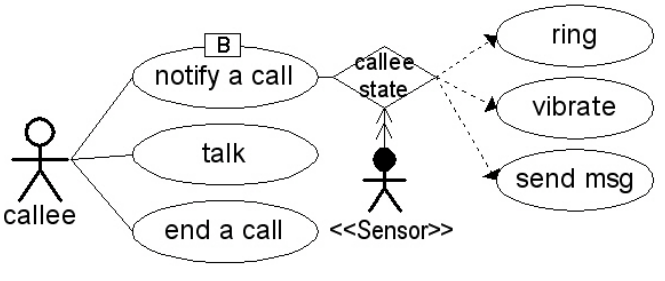


Fig. 4. Context-Aware Use Case Diagram for Smart Phone

Context-aware mobile phone or smart phone may help a user by providing three different notification methods of phone call. First, in a normal case, it notifies him/her of a phone call by ringing or vibration. Second, if he/she moves or runs, it notifies him/her by only ringing. And finally if he/she is busy, it refuses the phone call, and it sends an SMS message back to the caller. Even these simple scenarios make the system complex, and they must be described in use case explicitly in order to correctly

understand the system. Our context-aware use case diagram can meet this requirement as illustrated in Fig. 4. It shows basic business logic, context-aware service related use cases, context service type, and context type.

4.3 Context-Switch Diagram

During the creation of context-aware use case and use case diagram, stakeholders identify context-aware services, context type, and contexts. Then we can make a context-switch diagram. This diagram is the combination of class diagram and statechart diagram in UML, and it shows the relationship between context type and context, the relationship between context and context implementation, and the transition from one context to another context.

During the process specified in section 4.1, we can determine context set, C, for the smart phone: C= {normal, moving, busy}. The normal context means that a user is not busy, he/she can hear the phone ring, and he/she can feel the phone call vibration. The moving context is that as the user moves fast, he/she cannot feel the phone call vibration. The busy context means that the user cannot answer the phone because of a meeting or another busy activity.

We need to specify contexts, context type, and the change or switch of contexts to understand context-aware systems correctly. We specify them in our context-switch diagram. Fig. 5 shows a context-switch diagram for the smart phone. It illustrates not only the relationship between context type and contexts but also the relationship between the context and its implementation. Furthermore it shows the switch among the contexts. External or internal changes cause the context switch, and this diagram shows the change explicitly.

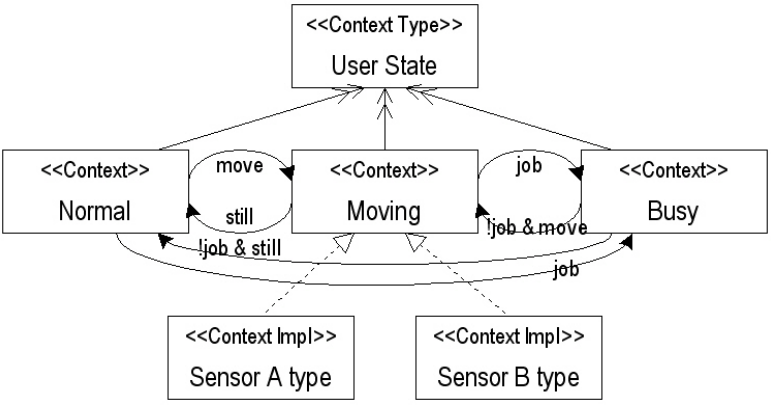


Fig. 5. Context-switch Diagram for Smart Phone

4.4 Dynamic Service Model

Context and context-aware service types help stakeholders to understand the systems, but they are not sufficient in themselves because they show only static aspects of the

systems. Stakeholders must understand the dynamic behavior of the system, in particular, the context-aware services.

The main parts in the system are context and context-aware service types as determined by context. Also, a system consists of several subsystems, so context data flow between the subsystems is also important. The dynamic model is described with context data flow between subsystems and the context-aware services. In general, the dynamic service model may be diagrammed by some alternation of a collaboration diagram or sequence diagram in UML.

Fig. 6 shows an example of a dynamic service model. It shows sub-modules of the smart phone and the interaction among them. First, the base station sends the call signal to the phone. Then the phone sends a request asking whether the user is busy to the schedule agent module, and the agent responds to the question. If he/she is busy, the Busy context is chosen. If he/she is not busy, the phone sends a request to the accelerometer to check the user's movement. If he/she is moving quickly, the Move context is chosen.

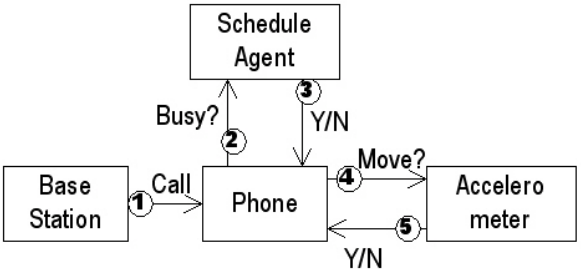


Fig. 6. Dynamic Service Model for Smart Phone

5 Conclusions

In a ubiquitous computing environment, the importance of context-aware services increases. Following this trend, there has been a great deal of research on context and context-aware systems. However, there has been little attention paid to requirement analysis methods for context-aware systems.

In this paper, we defined context as abstract and evaluative form and classified context-aware services into five types. Context and context-aware service types are very important for stakeholders in understanding and analyzing context-aware systems. Our context definition is abstract, so it is suitable for high-level presentation. Furthermore, it is also concrete enough to be described or bridged with specific values and specific external sensors. Our service type classification is comprehensive and easy to understand.

We also introduced a process for requirement analysis, context-aware user case diagram, context-switch diagram, and dynamic service model for context-aware systems. To reduce the complexity of context-aware systems, we separate concerns by splitting business logic and context logic in the requirement analysis phase.

Context-aware use case, context-switch diagram, and dynamic service model all help stakeholders to understand the systems, and they also provide a common basis for communication.

Until now we have studied the requirement phase, and in the future we will focus our effort into the development and programming method for context-aware systems. In particular, we have interests in system architecture and context-aware programming models.

References

1. Coutaz, J., Crowley, J.L., Dobson, S., Garlan, D.: Context is Key. CACM 48(3), 49–53 (2005)
2. Lieberman, H., Selker, T.: Out of context: Computer systems that adapt to, and learn from, context. IBM System Journal 39, IBM, 617–632 (2000)
3. Loke, S.: Context-Aware Pervasive Systems, Auerbach Pub. (2007)
4. Dey, A.K., Abowd, G.D.: Towards a Better Understanding of Context and Context-Awareness. In: Workshop on The What, Who, Where, When, and How of Context-Awareness in CHI'00 (2000)
5. Brooks, K.: The Context Quintet: Narrative Elements Applied to Context Awareness. In: HCI International Proc., pp. 1213–1217 (2003)
6. Schmidt, A., et al.: Advanced Interaction in Context. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, pp. 89–101. Springer, Heidelberg (1999)
7. Cheverst, K., et al.: Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences. CHI Letter 2(1), 17–24 (2000)
8. Schilit, B., Adams, N., Want, R.: Context-Aware Computing Applications. In: Workshop on Mobile Computing Systems and Applications, pp. 85–90 (1994)
9. Omasreiter, H., Metzker, E.: A Context-Driven Use Case Creation Process for Specifying Automotive Driver Assistance Systems. In: IEEE International Requirements Engineering Conf., pp. 334–339. IEEE, Los Alamitos (2004)
10. Mazuryk, L.K., Poulisse, G.J., van Eck, P.: Requirements Engineering for Pervasive Services. In: OOPSLA-workshop on Building Software for Pervasive Computing (2005)
11. Wagelaar, D.: Towards a Context-Driven Development Framework for Ambient Intelligence. In: Proc. of the Fourth IEEE International Workshop on Distributed Auto-adaptive and Reconfigurable Systems, pp. 304–309 (2004)
12. Lopes, A., Fiadeiro, J.L.: Context-Awareness in Software Architectures
13. Siewiorek, D., et al.: SenSay: A Context-Aware Mobile Phone. In: Proc. of International Symposium on Wearable Computers, pp. 248–249 (2003)

Performance Analysis of Child/Descendant Queries in an XML-Enabled Database

Eric Pardede[1], J. Wenny Rahayu[1], David Taniar[2], and Ramanpreet Kaur Aujla[1]

[1] Department of Computer Science and Computer Engineeering,
LaTrobe University, Bundoora 3086, Australia
{E.Pardede@,W.Rahayu@,xramanpreetkaur@students.}latrobe.edu.au
[2] Clayton School of IT, Monash University, Clayton 3800, Australia
David.Taniar@infotech.monash.edu.au

Abstract. The fact that XML data can be stored in different repositories creates a layer of complexity in efficient XML query processing. Most existing works discussed the query analysis for Native XML Database using XML Query Language such as XQuery and XPath. However, it is inevitable that many XML users still use established database repositories, such as relational and object-relational database, for their XML storage. These databases are recognized as XML-Enabled Database. The query analysis in this database family has not been well-researched. In this paper, we will perform a research on performance of child/descendant XML queries when the data is stored in XML-Enabled Database. For experimentation, we use Oracle DBMS, which has implemented different XML query languages, based on SQL/XML standard.

1 Introduction

The reasons for storing XML in a database system are the same as for relational data: persistent storage, transactional consistency, recoverability, high availability, security, efficient search and update operations, and scalability [4]. For this purpose, two databases are used, *Native XML Database* (NXD) and *XML Enabled Database* (XED)

Despite the claims of their powerful features, NXD is still in the development stage. Many people are reluctant to use this database family. On the other hand, XED is extended from established databases such as relational, object-oriented and object-relational, for XML data management purposes. Therefore, this database family can be used to store traditional data, such as relational data as well as XML data. It gives them an advantage over NXD.

Since XED is developed from another database, they support several query languages. XED that has relational foundation will support SQL as well as XML query languages such as XPath and XQuery. This fact has made the query performance analysis in XED unique. The analysis has to consider the nature of XML tree and also the relational-based performance measurement.

The purpose of this paper is to compare and analyze the performance of different querying techniques and to propose a recommendation of which query writing is suitable for one type of XML query, *Child/Descendant Query*. This work is motivated by the fact that current XEDs deal with considerably large number of queries and

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 749–762, 2007.
© Springer-Verlag Berlin Heidelberg 2007

thus, the most efficient process is required. For implementation, we use one major XED products, Oracle 10g Release 2 [5].

The comparative study is done by restructuring the query in terms of minimum resource utilization and query execution time. The comparison will consider the structure of the XML data in two aspects, the path and the level of hierarchy. The path analysis is a based on the way of writing XML tree path, and the level of hierarchy analysis is based on the placing selection and the projection at different levels of hierarchy.

2 Background

We start by briefly explain Child/Descendant XML query, which is the main topic of discussion in this paper and followed by different query structures in XED.

2.1 Child/Descendant XML Query

Based on the retrieval target, we classify XML Query into three types: *Child/Descendant (C/D) Queries*, *Parent Queries* and *Ancestor Queries* (see figure 1). In this paper, we focus on the first query type only. C/D queries are further classified into *C/D Sub-tree Queries*, *C/D Text Node Queries* and *C/D Attribute Queries*.

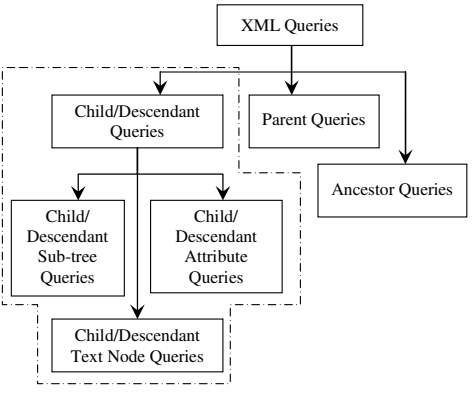


Fig. 1. XML-Enabled Database Query Classification

The main idea behind combining child and descendant queries is to check the performance difference for selection at the different levels of hierarchy while fixing the projection at a single node. For this purpose, the same element node will be retrieved in these queries to check the performance difference between the selections at the parent, ancestor or sibling nodes.

The query implementation is dependent on the database type. Even in one database family, such as XED, the implementation can be different from one product to another. We discuss about the variation in the next section.

2.2 Queries in XED

This section will show how an XED, in this case Oracle DBMS, retrieve XML documents. XML documents can be stored in Oracle databases using *CLOB*, *XMLType tables* or tables with *XMLType columns* [5]. The first type is used to store non-schema based XML documents, while the next two can be used to store schema-based XML documents.

Different ways of storing XML documents in XEDs introduce different query types. In [5], XML queries can be broadly described as: *Queries having extract, Queries having XMLTABLE, Queries having XMLQuery* and *Queries having XMLSequence*. The basic difference between these types is the structure used for projection and selection condition.

XPath Extract Query. It takes XPath expression as the projection and the selection elements. Function *existsNode* checks the XPath expression and returns true or false depending upon whether or not the XML document contains an XPath expression node. If the function returns true, the XML are retrieved using the extract function.

```
General Syntax:

  SELECT extract(<alias>.OBJECT_VALUE, '//<projection_element>')
  FROM <table_name> <alias>
  WHERE existsNode(<alias>.OBJECT_VALUE, '//<selection_condition>') = 1;
```

XMLTABLE XQuery. This query takes XQuery expression. containing the *FLOWR* expressions, as input. The "xtab" serves as a pointer which stores the results of the XQuery expressions and the results are displayed using "xtab.COLUMN_VALUE".

```
General Syntax:

  SELECT xtab.COLUMN_VALUE
  FROM <table_name>, XMLTABLE('<XQuery Expression using FLOWR Expressions>
      for $a in //<element_node>
        where <selection_predicate [conditional_expression]>
      return $a/<projection_element>' PASSING OBJECT_VALUE)xtab;
```

XML Query XQuery. This function takes XQuery expressions as input. There are two ways of using XMLQuery function. The syntax variation introduces the variation in the query execution.

```
General Syntax:

  SELECT XMLQuery('<XQuery Expression using FLOWR Expressions>
    for $a in //<element_node>
      where <selection_predicate [conditional_expression]>
      return $a/<projection_element>'
    PASSING OBJECT_VALUE RETURNING CONTENT) FROM <table_name>;

  SELECT XMLQuery('<XQuery Expression using FLOR Expressions>
    for $a in //<element_node>
      return $a/<projection_element>'
    PASSING OBJECT_VALUE RETURNING CONTENT) FROM <table_name>
  WHERE existsNode(<alias>.OBJECT_VALUE, '//<selection_condition>') = 1;
```

XPath XMLSequence Query. This query takes XPath expression as input. The *existsNode* function checks the XPath expression and returns true or false depending upon whether XML document contains an XPath expression node or not. XMLSequence function creates a collection of all these XML documents. The table function un-nests the collection and stores them in virtual table. Then *extractValue* function is used to extract the scalar values for projected element. If *existsNode* function returns false, then no rows are projected.

```
General Syntax:

  SELECT extractValue(value(x), '/<projection_element>')
  FROM <table_name> <alias>,table(
    XMLSequence(extract(<alias>.OBJECT_VALUE,'//<projection_element>')))x
  WHERE existsNode(<alias>.OBJECT_VALUE,'//<selection_condition>')= 1;
```

3 XML Query Optimization

As it is mentioned earlier, there are two major repositories for XML Database, NXD and XED. For NXD, there are works that optimized the XML query by extending the XML data model representation [3], by proposing new algorithms for query tree processing [8, 6], and by using optimized path expressions [1]. There are rooms for more research for this storage option. However, in this paper we focus on the query optimization in the other XML storage, XED.

3.1 Existing Works

Some of the existing works used approaches that are based on how the XML is stored in XED. For example, [10] investigated the query optimization using loosely-coupled information retrieval engine and using DBMS tables with query execution engines. It shows the effect of join algorithm and hardware cache utilization towards containment queries performance. Unfortunately, there is no detail discussion on how to improve the performance, so that, the containment queries can be efficiently processed. In this paper, we use different approach. We will only concern on the usage of DBMS tables and query execution engine for query optimization. However, we will show how to improve the query performance for different query types.

Other works focused on investigating translation from XML standard to XED language. [9] proposed an algebraic approach for order-sensitive XQuery processing over relational databases, which is called XQuery-to-SQL Order-sensitive Translation (XSOT). This work has more focus on direct rewriting from one query family to another query family. [11] investigated the algorithm using XML Algebra Tree (XAT) for explicitly represents the semantics of XQuery. Unfortunately, it does not clearly describe how it is applied in XED language such as SQL or SQL/XML. A similar approach is taken by [2], where the authors proposed a technique to rewrite XQuery by using an XML element constructor. It provided XQuery samples and their optimized result. However, there is no justification for the performance evaluation.

Table 1 summarizes the optimization techniques in XEDs with comments on their limitations. Another observation, these works focus on algorithm level and thus, do not give end users ability to perform query optimization.

Table 1. Optimization Techniques in XEDs

Technique	Ref	Comments
Join Algorithm and Cache Utilization	[10]	Missing discussion on performance improvement
XAT Query Rewriting and Cleanup	[11]	Missing description on implementation
Required Kinds of Query Rewriting	[2]	Missing performance evaluation
XSOT Approach	[9]	Missing description on implementation

3.2 Current Trends

Most XEDs are developed with relational database as the underlying DBMS. Among the products are Oracle, DB2, SQL Server, Cache and Matisse. In regards to query optimization, most of the products have been working in the following area.

- **XML Parsing Optimization.** It eliminates the need to keep re-parsing the XML documents. Any changes to the contents of a schema or Document Object Model (DOM) results in just writing the updated data back to the disk.
- **Node Searching Optimization.** A more optimum way to search a particular node can be investigated, instead of just using string comparisons. For example, Oracle [5] uses dynamic hash tables rather than string comparisons to perform node searches.
- **XML Schema Optimization.** Once the schema has been registered with the database, no additional schema parsing or validation is required before using it. If the DBMS requires information contained in the schema, it can access the required information directly from the cache.
- **Optimization of SQL functions that modify XML.** SQL/XML provides some functionality to update XML data, such as *updateXML*, *insertChildXML*, etc.
- **Rewriting of XPath expression.** XPath expressions can be rewritten to purely relational SQL statements which results in efficient processing. This rewritten SQL statement is processed by a database optimizer in the same manner as other SQL statements and the execution plan is based on a conventional relational algebra.

All these techniques, except for the last point, are based on hardware implementation and there has not been much work done for query optimization. As evident, query optimization in XED, is in infancy and has not been fully explored. One probable reason is the immaturity of XED standard by ISO/ANSI and W3C.

4 Taxonomy on Child/Descendant Query

C/D queries are further classified into three types: C/D Sub-tree queries, C/D Text Node queries and C/D Attribute queries. For the rest of the paper, we will refer them as sub-tree, text node and attribute query. This sub-division is based on the projection target. Sub-tree queries retrieve whole C/D nodes including text node and attributes associated with it. Text Node queries and Attribute queries retrieve just the text node of an element and attribute respectively.

4.1 Sub-tree Queries

These are the queries which involve the retrieval of C/D element(s) based on the selection on the parent, ancestor or sibling element. For example, Figure 2 shows a tree representation for C/D retrieval (node J), having selection on the parent element (node H), ancestor element (node B) and sibling element (node K). From the figure, we can see that this query type retrieve node J and its sub-elements. All these queries represented as tree structure can be written using XPath extract Query, XMLTABLE XQuery and XMLQuery XQuery syntax.

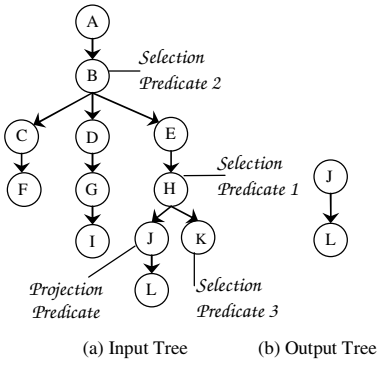


Fig. 2. Tree Repr. for Sub-tree Queries

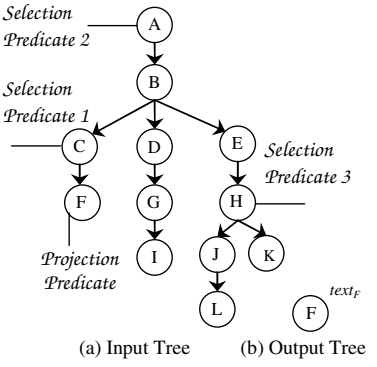


Fig. 3. Tree Repr. for Text Node Queries

4.2 Text Node Queries

These queries involve with the projection of a text node associated with the C/D element based on the selection on the parent, ancestor or sibling element. For example, Figure 3 shows a tree representation for the C/D Text Node retrieval (element F) having selection on parent element (node C), ancestor element (node A) and sibling element (node H).

From the figure, the output of the query is the text associated with element F. In XED, these queries can be performed using SQL function `text()` in XPath XMLSequence Query and XPath extract Query. The changed syntax of XPath extract Query for Text Node query is given below:

```
SELECT extract(<alias>.OBJECT_VALUE,'//<projection_element>/text ( )')
FROM <table_name> <alias>
WHERE existsNode(<alias>.OBJECT_VALUE,      '//<selection_condition>');
```

4.3 Attribute Queries

These queries involve in the projection of attributes associated with the C/D element based on the selection on the parent, ancestor or sibling element. For example, Figure 4 shows a tree representation for the C/D Attribute retrieval (node H) having selection on parent element (node E), ancestor element (node B) and sibling element (node F).

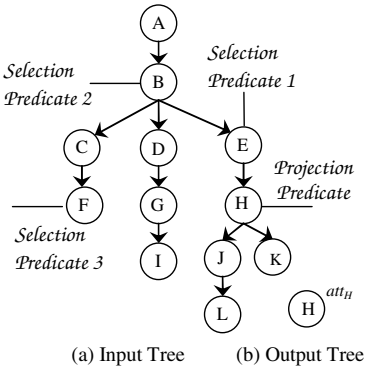


Fig. 4. Tree Representation for Attribute Queries

From the figure, the output of the query is the attribute associated with node H. Attribute queries use XPath extract Query, XMLTABLE XQuery and XMLQuery XQuery with a little modification. The changed syntax of above mentioned queries in order to retrieve attribute associated with C/D elements is given below.

```
XPath extract attribute Query:

SELECT extract(<alias>.OBJECT_VALUE,
   '/<element_having_attribute>/@<projection_attribute>')
FROM <table_name> <alias>
WHERE existsNode(<alias>.OBJECT_VALUE, '//<selection_condition>');

XMLTABLE attribute XQuery:

SELECT xtab.COLUMN_VALUE
FROM <table_name>, XMLTABLE('<XQuery Expression using FLOWR Expressions>
   for $a in //<element_node> where
   <selection_predicate[conditional_expression]>
   RETURN $a/<element_having_attribute>/@<projection_attribute>'
   PASSING OBJECT_VALUE)xtab;

XMLQuery attribute XQuery:

SELECT XMLQuery('<XQuery Expression using FLOWR Expressions>
   for $a in //<element_node> where
   <selection_predicate[conditional_expression]>
   RETURN $a/<element_having_attribute>/@<projection_attribute>'
   PASSING OBJECT_VALUE RETURNING CONTENT)
FROM <table_name>;
```

5 Queries Performance Experiments

5.1 Experimental Setup and Tool

For all practical purposes, Oracle 10g Release 2 [5] was used for this paper. The experiments were performed on a Windows workstation having 40 GB of disk storage and 504MB of memory. XML Schema was created based on a publicly available

XML document (http://www.dia.uniroma3.it/Araneus/Sigmod/Record/SigmodRecord/SigmodRecord.xml)

The size of the data used for the experiment was 1.01MB. The table was created based on the XML Schema. The XML documents stored contain not only element nodes but also attribute and text nodes. TKPROF utility and Timing utility [5] were used to do performance analysis. TKPROF utility [7], also known as SQL Trace, reports information about each SQL statement executed with the resources it has used, the number of times it was called, and the number of rows which it processed.

Performance analysis of the queries involves the comparison of the statistical information of the queries. The information is based on the following parameters.

- **CPU Time.** It refers to time taken to parse, execute, or fetch calls for the statement. Query performance is shown by Execute and Fetch phases
- **TKPROF Elapsed Time.** It refers to the time required for all parsing, executing, or fetching calls for the statement. This is the total time from start to finish of the execution of the query
- **Timing Elapsed Time.** It is the total execution time of the query obtained by timing utility of SQL*Plus.

5.2 Experimentation

Sub-tree Queries Experimentation. *"Retrieve title C/D element when the selection is on parent/ancestor/sibling element."*

The query for this problem can be categorized into six, based on different combinations of projections and selections in the hierarchy (see table 2). The projection predicate is fixed at *title* element but the selection predicate is located at different levels of hierarchy. Each category can have four ways of query writing.

Table 2. Sub-Tree Query Categories

Cat	Description
1	Selection pred. at top level ancestor element
2	Selection pred. at first level ancestor element
3	Selection pred. at parent element
4	Selection pred. at child elements of top level ancestor element
5	Selection pred. at lowest level of hierarchy
6	Selection pred. on a sibling element

Figure 5 represents the selection and the projection predicates for category 2. In this case, we *retrieve title descendant element when attribute textValue1 (ancestor element) is 50*. The projection predicate is *title* and the selection predicate is *articles* (textValue1 attribute) located at the fourth level of hierarchy (see Figure 5). Table 3 represents the possible structures for this query.

This query can be written in three different ways based on sub-section 2.2. Queries 1 and 4 are written using XPath extract Query. Query 4 is written using the same query structure as Query 1. However, the way it is written is different. Query 2 is written using XMLTABLE XQuery. Query 3 is written using XMLQuery XQuery.

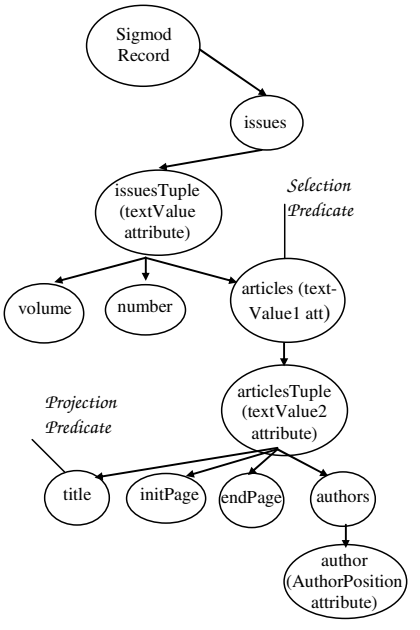


Fig. 5. Sub-Tree Query Tree Representation

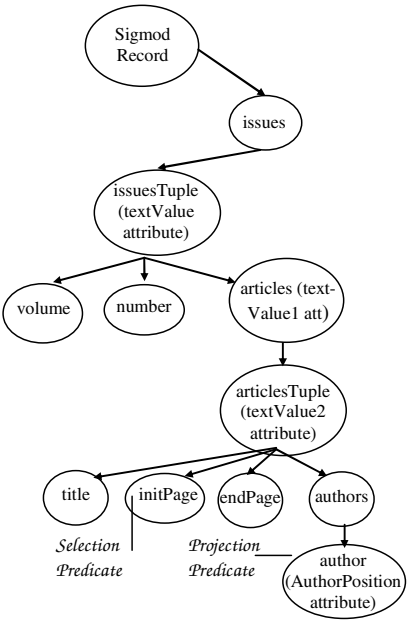


Fig. 6. Text Node Query Tree Representation

Table 3. Possible C/D Query Structures for the Problem Definition

Q	Query Structures associated with Sub-Tree Query Category2

1.	`SELECT extract(s.OBJECT_VALUE, '//title')"TITLE"`
	`FROM SIGMOD s`
	`WHERE existsNode(s.OBJECT_VALUE,'//articles[@textValue1="50"]')= 1;`

2.	`SELECT xtab.COLUMN_VALUE "TITLE"`
	`FROM SIGMOD, XMLTABLE ('for $a in //articles let $b:=$a[@textValue1="50"]`
	`return $b//title' PASSING OBJECT_VALUE)xtab;`

3.	`SELECT XMLQuery('for $a in //articles return $a//title'`
	`PASSING OBJECT_VALUE RETURNING CONTENT)"TITLE" FROM SIGMOD s`
	`WHERE existsNode(s.OBJECT_VALUE,'//articles[@textValue1="50"]')= 1;`

4.	`SELECT extract(s.OBJECT_VALUE, '/SigmodRecord/*//articles/*//title') "TITLE"`
	`FROM SIGMOD s`
	`WHERE existsNode(s.OBJECT_VALUE,'/SigmodRecord/*//articles`
	`[@textValue1="50"]')= 1;`

Text Node Queries Experimentation. *"Retrieve Text Node associated with author/volume C/D element when the selection is on a Parent/Ancestor/Sibling element."*

The Text Node Queries are based on projection of two different elements. One of which is the topmost element (*volume*) and the other is the bottom-most element (*author*). Based on the projection and the selection nodes, we can classify this query into 12 categories shown in table 4. There are 37 different ways of writing queries for these categories.

Figure 6 represents selection and projection predicate for query category 5. In this case, we *retrieve text node of author when element initPage is 114 or 144.* The projection predicate for this query is *author* element and the selection predicate is

Table 4. Text Node Query Categories

Cat	Description
1	Selection pred. at top level ancestor element & projection pred. at bottom-most descendant element
2	Selection pred. at 2nd level ancestor element & projection pred. at bottom-most descendant element
3	Selection pred. at child element of top level ancestor element & projection pred. at bottom-most descendant element
4	Selection pred. on the attribute associated with projection pred. element & projection pred. at bottom-most element
5	Selection pred. on element which is a sibling to the parent element of projection pred. element & projection pred. at bottom-most element
6	Selection pred. at 1st level ancestor element & projection pred. at bottom-most desc. element
7	Selection pred. at parent element & projection pred. at top-most child element
8	Selection pred. at attribute associated with a sibling element & projection pred. at top-most child element
9	Selection pred. at 1st level descendant element of parent element & projection pred. at top-most child element
10	Selection pred. on a sibling element & projection pred. at top-most child element
11	Selection pred. at lowest level of hierarchy & projection pred. at top-most child element
12	Selection pred. at 2nd level descendant element of parent element & projection pred. at top-most child element

Table 5. Possible Text Node Query Structures for the Problem Definition

Q	Query Structures associated with Text Node Query Category 5

5.	`SELECT extract(s.OBJECT_VALUE, '//author/text()') "AUTHORS"`
	`FROM SIGMOD s`
	`WHERE existsNode(s.OBJECT_VALUE,'//articlesTuple[initPage="144" or initPage="114"]')= 1;`

6.	`SELECT extractValue(value(AUTHORS), '/author')"AUTHORS"`
	`FROM SIGMOD s, table(XMLSequence(extract(s.OBJECT_VALUE,'//author')))AUTHORS`
	`WHERE existsNode(s.OBJECT_VALUE,'//articlesTuple[initPage="144" or initPage="114"]')= 1;`

7.	`SELECT extract(s.OBJECT_VALUE,`
	`'/SigmodRecord/*//articlesTuple/*//author/text()') "AUTHORS" FROM SIGMOD s`
	`WHERE existsNode(s.OBJECT_VALUE,'//articlesTuple[initPage="144" or initPage="114"]')= 1;`

initPage element located at sixth level of hierarchy. Table 5 represents the possible query structures for this category.

This query can be written in different ways based on the query structures discussed in section 2.2. Query 5 and Query 7 are written using XPath extract Query with `text()` function. Query 7 is written using the same query structure as Query 5 but the path in Query 7 is written in different way. Query 6 is written using XPath XMLSequence Query. This query takes XPath expression as input.

Attribute Queries Experimentation. *"Retrieve Attributes associated with C/D element when the selection is on a parent/ancestor element."*

The attribute queries are based on projection of three different C/D attributes: textValue2, AuthorPosition, textValue1 (see Figure 7). Based on the projection and the selection nodes, we can classify this query into 6 categories shown in table 6. There are 24 different ways of writing queries for these categories.

Table 6. Attribute Query Categories

Cat	Description
1	Selection pred. at top level ancestor element & projection pred. at middle level of hierarchy
2	Selection pred. at parent element & projection pred. at middle level of hierarchy
3	Selection pred. at top ancestor element & projection pred. at lowest level of hierarchy
4	Selection pred. at 2nd level ancestor element & projection pred. at lowest level of hierarchy
5	Selection pred. at 1st level ancestor element & projection pred. at lowest level of hierarchy
6	Selection pred. at parent element & projection pred. at top level of hierarchy

Figure 7 represents the selection and projection predicates for Attribute Query category 5. In this case, we *retrieve the AuthorPosition descendant attribute when the attribute textValue2 (ancestor element) is 51 or 52.*

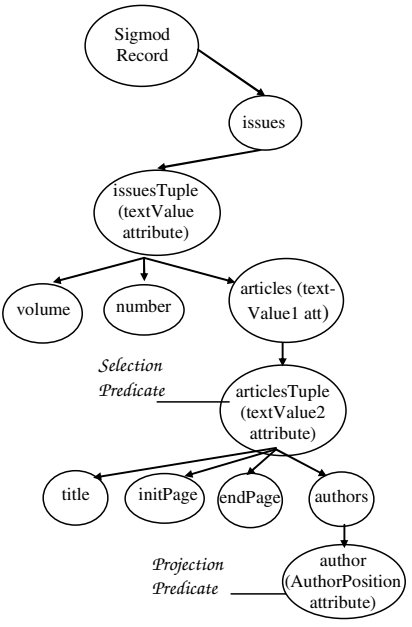


Fig. 7. Attribute Queries Tree Representation

Table 7 represents possible query structures. Query 8 and 11 are written using XPath extract attribute query. Query 9 is written using XMLTABLE attribute XQuery. Query 10 is written using XMLQuery attribute XQuery.

5.3 Experimental Results

We provide three different categories from three different query structures. They are category 1, category 5 and category 5 respectively for Sub-Tree query, Text Node query and Attribute Query.

Table 7. Possible Attribute Query Structures for the Problem Definition

Q	Query Structures associated with Text Node Query Category 5
8.	SELECT extract(s.OBJECT_VALUE, '//@AuthorPosition') "AUTHORPOSITION" FROM SIGMOD s WHERE existsNode(s.OBJECT_VALUE,'//articlesTuple[@textValue2="51" or @textValue2="52"]')= 1;
9.	SELECT xtab.COLUMN_VALUE "AUTHORPOSITION" FROM SIGMOD, XMLTABLE ('for $a in //articlesTuple where $a/@textValue2 = "51" or $a/@textValue2 = "52" return $a//@AuthorPosition' PASSING OBJECT_VALUE)xtab;
10.	SELECT XMLQuery('for $a in //articlesTuple return $a//@AuthorPosition' PASSING OBJECT_VALUE RETURNING CONTENT)"AUTHORPOSITION" FROM SIGMOD s WHERE existsNode(s.OBJECT_VALUE,'//articlesTuple[@textValue2="51" or @textValue2="52"]')= 1;
11.	SELECT extract(s.OBJECT_VALUE, '/SigmodRecord/*//articlesTuple/*// @AuthorPosition') "AUTHORPOSITION" FROM SIGMOD s WHERE existsNode(s.OBJECT_VALUE,'/SigmodRecord/*//articlesTuple [@textValue2="51" or @textValue2="52"]')= 1;

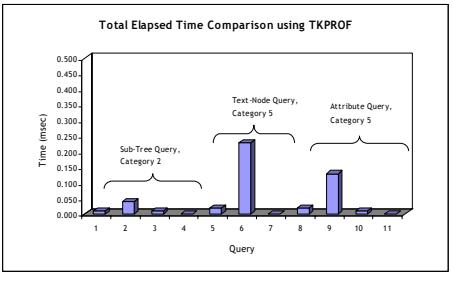


Fig. 8. TKPROF Total Elapsed Time

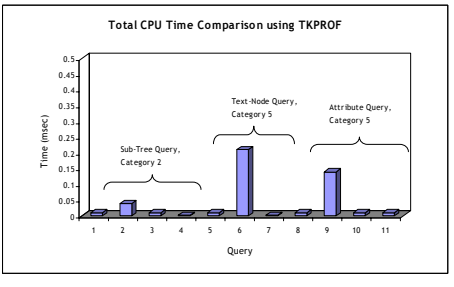


Fig. 9. TKPROF Total CPU Time

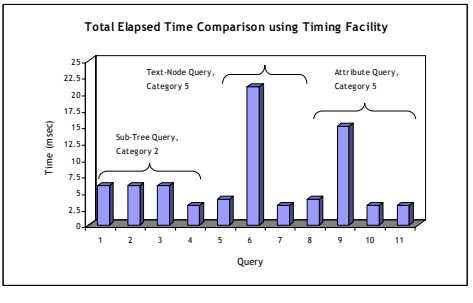


Fig. 10. Total Elapsed Time Comparison using Timing Facility

Each of the categories can be written in different ways. There are 4, 3 and 4 ways of writing queries in the respective categories as it is shown in tables 3, 5 and 7. The analysis of the Execution Plan of C/D Queries is explained below:

- **Type of operation.** All the execution plans consist of full table access. Execution plan for query 4, query 7 and query 11 are a subset of all other queries.

- **Number of Steps.** The execution plan of query 4, query 7 and query 11 involve only one step, fewer than for any other execution plan.
- **Number of rows.** In query 4, query 7 and query 11, only one row is being accessed and retrieved in the execution plan. It is less when compared with execution plans of other queries. The execution plan of these three queries involve full table access, whereas in others, several steps had to be performed in order to obtain the required output.

From the above analysis of the execution plan, it can be concluded that the execution plan for queries 4, 7 and 11 are the most cost effective for its respective query type and category. A comparative analysis of total elapsed time using TKPROF, Total CPU time and total elapsed time using DBMS timing property is graphically represented in figures 8 to 10. We can see how queries 4, 7 and 11 incur the smallest cost for each respected category in each comparative terms.

5.4 Experimental Summary

Based on the performance experimentations, we can derive some analysis for writing C/D XML Query in XED.

- From the graphs, we can see XQuery written in terms of XPath Expressions using *extract* SQL function is more cost effective than XQuery written in terms of FLOWR Expressions using *XMLTABLE* and *XMLQuery* SQL Functions. XPath extract query yields good performance even when the selection is at bottom of hierarchy.
- In all three query types, having the selection predicate at the top level of the hierarchy performs better than having the predicate at the bottom. The performance for the queries having selection at top or bottom can be performed by writing the path //element as */root_node_element/*//element* in both selection and projection.
- For all of the queries and different categories, XPath extract query performed better when path in the selection and projection predicates are written as */root_node_element/*//projected_or_selected_element*. The performance of the XMLTABLE XQuery and XMLQuery XQuery can also be improved by writing the path as the same way.

6 Conclusions and Future Work

In this paper we analyse various types of the C/D queries in an XED. Based on the result of the performance analysis, each type of query can be written in a most optimized query structure. We use four queries structures, namely XPath extract query, XMLTABLE XQuery, XMLQuery XQuery and XPath XMLSequence query.

The performance is then analysed by using the statistics generated by the XED. The most optimized query structure was identified in terms of cost associated with the execution of the query.

For future work, we can perform a performance analysis for different XML query such as parent and ancestor queries. A benchmark study of different XED can also be performed to help database users determine the storage option for their XML data.

References

1. Bruno, N., Koudas, N., Srivastava, D.: Holistic twig joins: optimal XML pattern matching. In: SIGMOD'02, ACM Press, New York (2002)
2. Grinev, M., Pleshachkov, P.: Rewriting-based Optimization for XQuery Transformational Queries. In: IDEAS'05, IEEE Computer Society Press, Los Alamitos (2005)
3. Jagadish, H.V., Lakshmanan, L.V.S., Scannapieco, M., Srivastava, D., Wiwatwattana, N.: Colorful XML: One Hierarchy Isn't Enough. In: SIGMOD'04, ACM Press, New York (2004)
4. Lapis, G.: XML and Relational Storage-Are they mutually exclusive. IBM Corporation (2005)
5. Oracle.: Introduction to Oracle XML DB. Oracle XML DB Developer's Guide, 10g Release 2. (2006b) Available at http://download-west.oracle.com/docs/cd/B19306_01/appdev.102/b14259/xdb01int.htm#i1047170
6. Paparizos, S., Wu, Y., Lakhsmanan, L.V.S., Jagadish, H.V.: Tree Logical Classes for Efficient Evaluation of XQuery. In: SIGMOD'04, ACM Press, New York (2004)
7. Srivastava, P.G.: Performance Evaluation Tools on Oracle, Department of Computer Science and Computer Engineering, La Trobe University (2002)
8. Wang, G., Liu, M., Yu, J.X., Sun, B., Yu, G., Lv, J., Lu, H.: Effective schema-based XML query optimization techniques. In: IDEAS'03, IEEE Computer Society Press, Los Alamitos (2003)
9. Wang, L., Wang, S., Murphy, B., Rundensteiner, E.: Order-sensitive XML Query Processing over Relational Sources: An Algebraic Approach. In: IDEAS'05, IEEE Computer Society Press, Los Alamitos (2005)
10. Zhang, C., Naughton, J., Dewitt, D., Luo, Q., Lohman, G.: OHMAN. On Supporting Containment Queries in Relational Database Management Systems. In: SIGMOD'01, ACM Press, New York (2001)
11. Zhang, X., Pielech, B., Rundesnteiner, E.: Honey, I Shrunk the XQuery! — An XML Algebra Optimization Approach. In: WIDM'02, ACM Press, New York (2002)

Diagonal Data Replication in Grid Environment

Rohaya Latip, Hamidah Ibrahim, Mohamed Othman, Md Nasir Sulaiman, and Azizol Abdullah

Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
{rohaya,hamidah,mothman,nasir,azizol}@fsktm.upm.edu.my

Abstract. Replication is a useful technique for distributed database systems and can be implemented in a grid computation environment to provide a high availability, fault tolerant and enhance the performance of the system. The performances addressed in this paper are data availability for read and write operations and communication cost, which are compared with the previous replication protocol, Three Dimensional Grid Structure (TDGS). This paper discusses a new protocol of replicating data for grid environment by putting the topology of the grid network in a logical grid structure and in a quorum where the selected replicas will be the primary database for replicating data to enhance the read and write availability of the data and to reduce the communication cost. The selection of replica is dependent on the diagonal location of the structured grid network. The middle replica is selected because it is easy to access and is the shortest path to get a copy of the data. To evaluate our model, we developed a simulation model in Java. Our results prove that by replicating diagonally in a quorum improves the performance of the data availability and communication cost in a large network size such as grid computational.

Keywords: Data replication, Grid, Data management, Simulation.

1 Introduction

A grid is a distributed network computing system, a virtual computer formed by a networked set of heterogeneous machines that agree to share their local resources with each other. A grid is a very large scale, generalized distributed network computing system that can scale to internet size environment with machines distributed across multiple organizations and administrative domains [1]. I. Foster [2] defines grid as technologies and infrastructure to support the sharing and coordinated use of diverse resources in dynamic, distributed virtual organizations which are the creation from geographically distributed components operated by distinct organization with differing policies of virtual computing system that are sufficiently integrated to deliver the desired QoS. Ensuring efficient access to such a huge network and widely distributed data is a challenge to those who design, maintain and manage the grid network. The availability of a certain data at a huge network and replicating data at a minimum communication cost is one of the issue. In our work, we investigate the use of replication on a grid network to improve its ability to access data efficiently.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 763–773, 2007.
© Springer-Verlag Berlin Heidelberg 2007

To address the availability, we replicate data in a logical structured grid topology network and select a replica from a diagonal location of the grid topology which has been organized in quorums. The middle replica of an odd number of nodes in the quorum is selected. This is because it is easy to access the database and it is usually the shortest path to the replicated data because of its location in the middle of the quorum. We use Java to run this replication protocol. The paper is organized as follow. Section 2, overview the previous works on grid replication and distributed database. In section 3, we introduce our replication protocol and its algorithm. Section 4 describes the simulation framework and Section 5 discusses the simulation results. Brief conclusions and future works are discussed in Section 6.

2 Related Work

There are some works that have been carried out in data replication for grid environment such as Dynamic Data Replication [3, 4, 5]. K. Ranganathan [3, 4] simulated using discrete event simulator, ChicagoSim, to evaluate the performance of different combinations job and task scheduling. Job scheduling and data movement (replication) algorithm evaluates various combinations. They suggested that while it is necessary to consider the impact of replication on the scheduling strategy, it is not always necessary to couple data movement and computational scheduling. Instead, they suggested these two activities can be addressed separately, thus significantly simplified the design and implementation of the overall Data Grid system.

[6, 7, 8] introduce a set of replication management services and protocol that offer high data availability, low bandwidth consumption, increased fault tolerant and improved scalability of overall system. Their protocol of replication is made based on a cost model that evaluates data access costs and performance gains of creating each replica. Running on a network simulator NS, called the GridNet, the estimation of costs and gains is based on such factors as run-time accumulated read/write statistics, response time, bandwidth and replica size. In this protocol, replicas are organized in a combination of hierarchal and flat topologies that represent propagation graphs that minimizes inter-replica communication costs.

M. Tang [9] introduce two dynamic replication algorithms, namely, Simple Bottom-Up (SBU) and Aggregate Bottom-Up (ABU), for the multi-tier Data Grid. Fig. 1 from [9] shows the architecture supporting dynamic replication. They developed a multi-tier Data Grid simulator called DRepSim for studying the performances of the dynamic replication algorithms. SBU creates replicas near to the clients that requested the data files based on the access history while, ABU aggregates the history records to the upper tier step by step till it reaches the root.

Their simulation results show that both algorithms can reduce the average response time of data access greatly compared to the static replication method. ABU can achieve great performance improvements for all access patterns even if the available storage size of the replication server is very small. Comparing the two algorithms to Fast Spread Dynamic Replication strategy, ABU proves to be superior. As for SBU, although the average response time of Fast Spread Dynamic Replication strategy is better in most cases, Fast Spread's replication frequency is too high to be applicable in the real world.

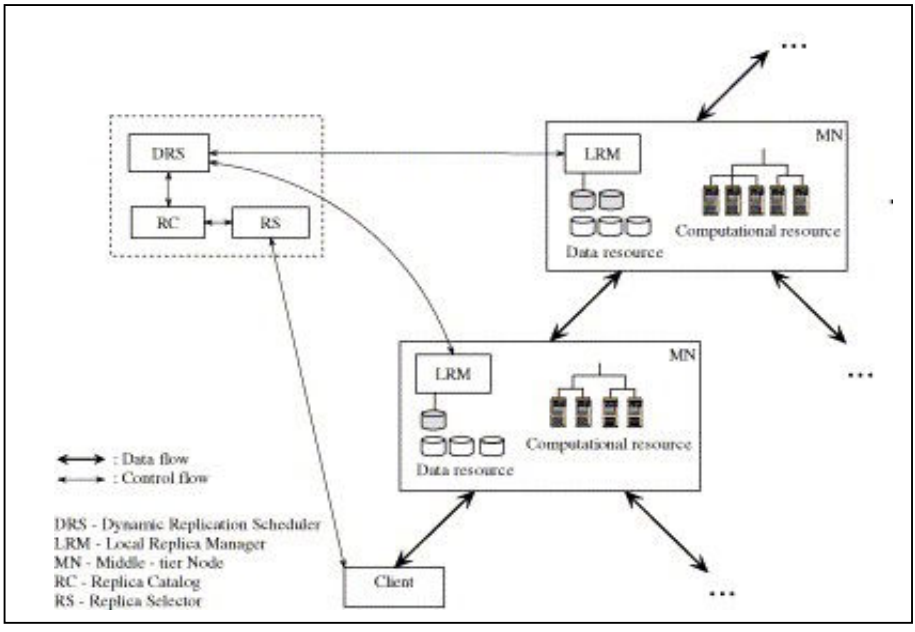


Fig. 1. System architecture supporting dynamic replication

Three Dimensional Grid Structure (TDGS) implemented by [10] is a protocol replicating its data or copies in logical box shape structure with four planes. Fig. 2. illustrates eight copies of data object.

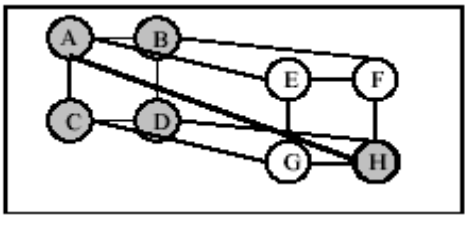


Fig. 2. Eight copies of data object

Read operations in TDGS are executed by acquiring a read quorum that consists of any hypotenuse copies. For example shown in Fig. 2, hypotenuse copies will be {A, H}, {B, G}, {C, F}, and {D, E}. Read operations are executable by these pairs of hypotenuse copies.

Write operations are executable from any planes that consist of hypotenuse copy. Planes in Fig. 2 are {H, A, B, C, D}, {C, F, E, G, H), and etc. Example, to execute read operation, copies from {A, H} must be accessible and to execute write operation, copies from {H, A, B, C, D} must also be accessible.

Read operations are executed by acquiring a read quorum, which must come in pairs at the vertices of the box shape structure if one of the copies of each pairs is unavailable, thus that hypotenuse copies are not accessible. Therefore, write operations are not executable at that write quorum. Read and write quorum must intersect; if that edge of read quorum is not accessible thus the write quorum is also not accessible to update the latest data. This will affect the consistency of the data.

Other limitation of this protocol is that, it must be formed in a perfect square box and if new copies are added, thus more copies are added at the other sides of the box to get a perfect square box. This somehow will increase the read and write quorum size and affect the communication cost.

3 Diagonal Data Replication Algorithms

In our method, all nodes are logically organized into two dimensional grid structure. We assume that the replica copies are in the form of data file and all nodes are operational meaning that the copy at that node is available. The data file will be replicated to only one node of the diagonal sites.

This protocol uses quorum to put nodes in cluster. Quorum is grouping the nodes or databases as shown in Fig. 3. Fig. 3 illustrated how the quorums for network size of 7 x 7 are grouped by nodes of 5 x 5 in each quorum. Nodes which are formed in quorum intersect with other quorums. This is to make sure that they can communicate or read other data from other nodes which is in another quorum.

The group of nodes called quorum, *R* must be odd so that the replica of the middle diagonal nodes can be selected such as the black circles in Fig. 4. For example in Fig. 4, we select the middle node which is node s(3,3) to have the copy of data.

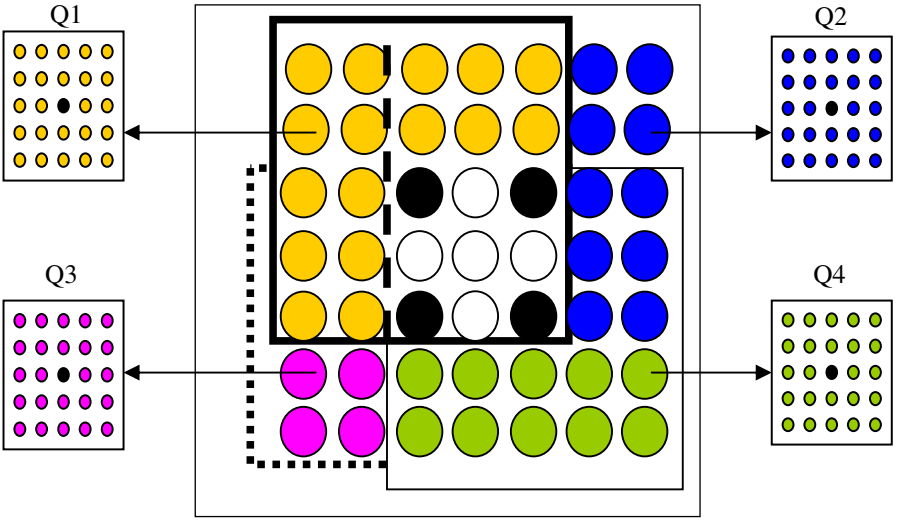


Fig. 3. Four quorums obtain in a 7 x 7 network size

Definition 3.1. Assume that a database system consists of n x n nodes that are logically organized in the form of two dimensional grid structure. All sites are labeled s(i,j), $1 \le i \le n$, $1 \le j \le n$. the diagonal site to s($i,,j$) is {s(k,l)| $k = i + 1$, $l = j + 1$, and $k,l \le$ n, if $i = n$ initialized $i = 0$, if $j = n$, initialized $j = 0$}. A diagonal set, D(s), is a set of diagonal sites, D(s) = $s(m,n)$, where $m = n$ and m, n = 1, 2,…, 5 as in Fig. 4. Fig. 4 has 25 copies where the size network is 5 x 5 nodes.

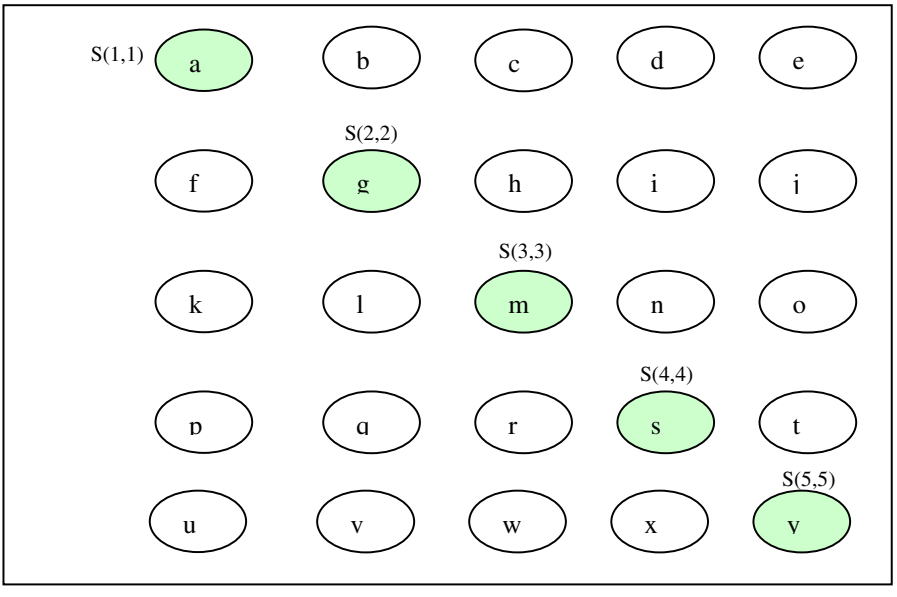


Fig. 4. A grid organization with 25 nodes, each of the node has a data file a,b,…, and y respectively

For example, from Fig. 4, the diagonal site D(s) is {s(1,1), s(2,2), s(3,3), s(4,4), s(5,5)} and the node to have the copy of data is the middle node s(3,3). Since the data file is replicated only on one node, thus it minimizes the number of database update operations. Its diagonal sites are assigned with vote one and vote zero otherwise. A vote assignment on grid, B, is a function such as,

$$B(s(i,j)) \in \{0, 1\},\ 1 \le i \le n,\ 1 \le j \le n$$

Where B(s(i,j)) is the vote assigned to site s(i,j). This assignment is treated as an allocation of replicated copies and a vote assigned to the site results in a copy allocated at the diagonal site. That is,

1 vote ≡ 1 copy.

$$\text{Let } L_B = \sum_{i=1}^{n} B(s(i,j))$$

Where L_B is the total number of votes assigned to primary node and its diagonal sites. Thus $L_B = 1$.

Let r and w denote the read quorum and write quorum respectively. To ensure that read operation always gets the updated data, $r + w$ must be greater than the total numbers of copies (votes) assigned to all sites. To make sure the consistency is obtained, the following conditions must be fulfilled [11].

i. $1 \leq r \leq L_B$, $1 \leq w \leq L_B$
ii. $r + w = L_{B+1}$.

These two conditions ensure that there is a nonempty intersection of copies between read and write quorum. Thus, the conditions ensure that a read operation can access the most recently updated copy of the replicated data.

Let $S(B)$ be the set of sites at which replicated copies are stored corresponding to the assignment B. Then,

$S(B) = \{s(i,j) | B(s(i,j)) = 1, 1 \leq i \leq n, 1 \leq j \leq n\}$

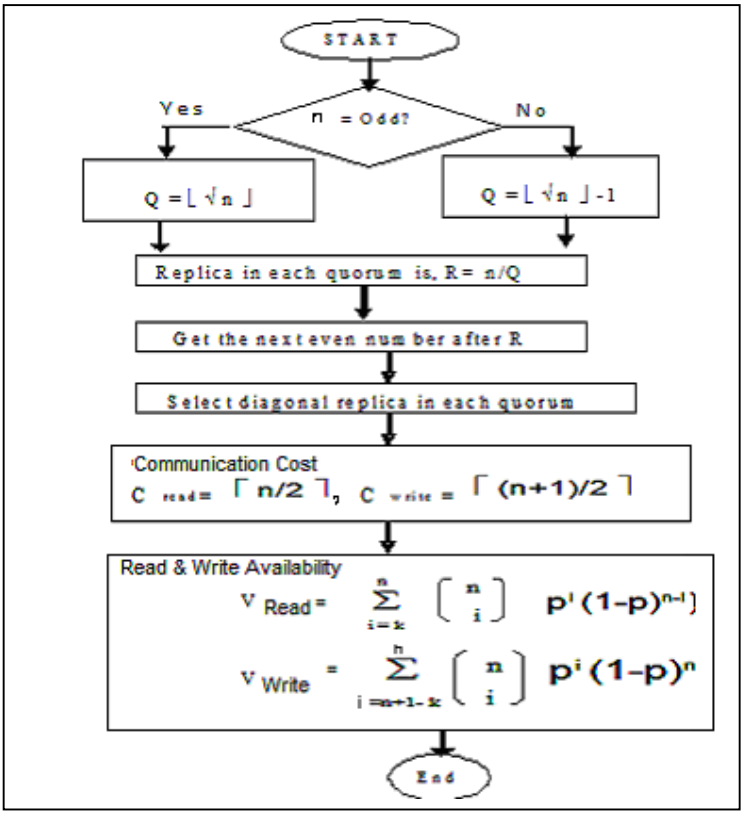


Fig. 5. Flow chart of the simulation design

Definition 3.2. For a quorum, *q*, a quorum group is any subset of S(B) where the size is greater than or equal to *q*. The collection of quorum group is defined as the quorum set.

Let Q(B,q) be the quorum set with respect to assignment *B* and quorum, *q*, then

$Q(B,q) = \{G | G \subseteq S(B) \text{ and } |G| \geq q\}$

For example, from Fig. 4, let site s(3,3) be the primary site of the master data file *m*. Consider an assignment B for the data file *m,* such that

$B_a(s(3,3)) = 1$ and $L_{Ba} = B_a(s(1,1)) = 1$.

Therefore, $S(B_a) = \{s(3,3)\}$.

This protocol is suitable for a huge network size. For example 27 x 27 nodes, where *n* is 27. Number of quorum, *Q* is 5 as $Q = \lfloor \sqrt{27} \rfloor$ then *R,* the number of nodes in each quorum is 7, where R = n/Q = 5.4 and the next odd number is 7. All quorum will intersect as define in Definition 3.1. Only the middle node of the diagonal node in D(s) will have the copy of data file as illustrated in Fig. 4.

4 Simulation Design

The model was developed using Java and limited to not more than 27 x 27 grid, which is 729 nodes. The design of this model is as in Fig. 5, where it shows the flow of the simulation.

5 Performance Analysis and Comparison

To analyze the performance of read and write communication cost and availability, Equation 1 and 2 is used from [12], where *n* is the grid column or row size, example *n* is 7, thus 7 x 7 nodes of grid is the network size and *p* is the probability of data available which is between 0 to 1. Whereas, *k* is the increment of *n*.

For communication cost of read operation is as in Equation 1 and communication cost of write operation is as in Equation 2.

$$C_{read} = \lceil n/2 \rceil \quad (1)$$

$$C_{write} = \lceil (n+1)/2 \rceil \quad (2)$$

From [12], read and write availability are as in Equation 3 and 4 respectively.

$$V_{Read} = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-} \quad (3)$$

$$V_{Write} = \sum_{n+1-k}^{n} \binom{n}{i} p^i (1-p)^{n-i}) \quad (4)$$

From Fig. 5, to get the number of quorums which are suitable for the size of the grid network. A bigger grid network will have more quorum. The number of columns and rows in each quorum must be odd, to get the middle replica. By selecting only one middle replica in each quorum will minimize the communication cost.

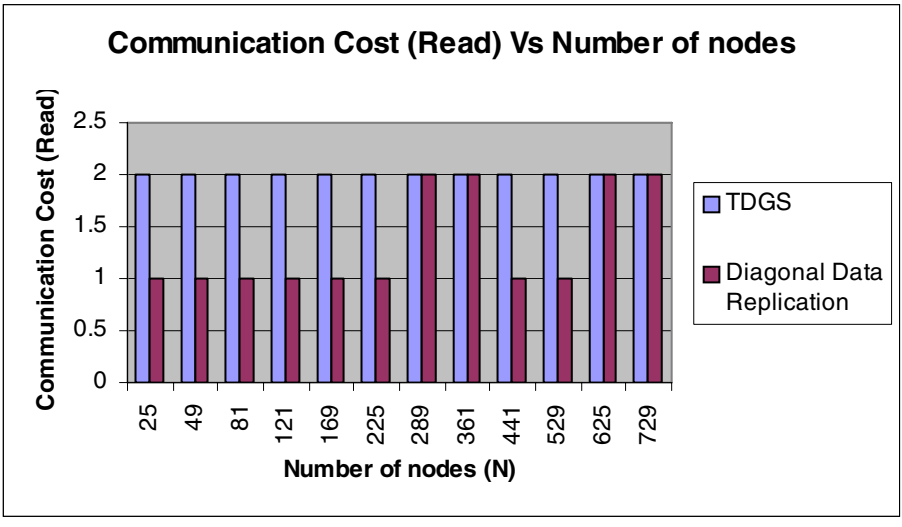


Fig. 6a. Communication cost for read operation

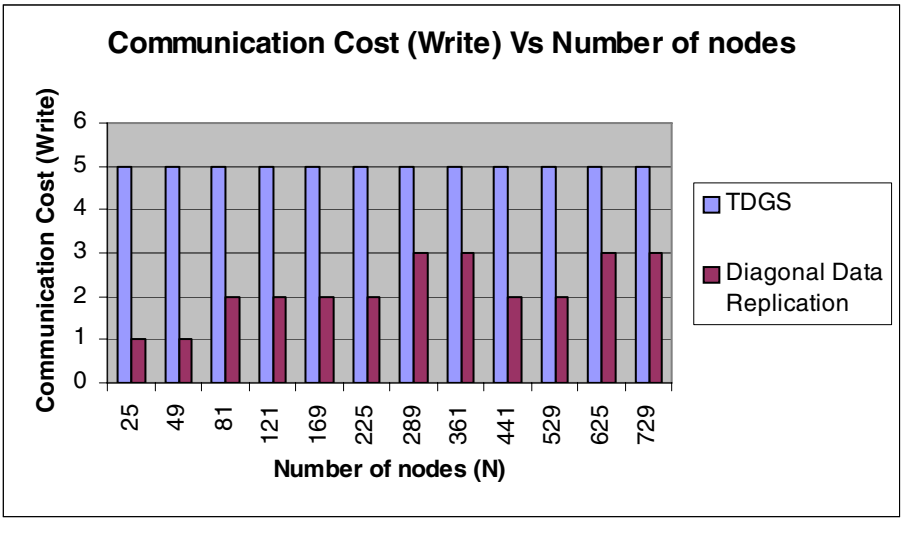


Fig. 6b. Communication cost for write operation

Fig. 6a, presents the communication cost for read operation of nodes from 25 to 729 nodes. Fig. 6a shows that the diagonal replication protocol has reduced the number of communication cost for read operation compare to TDGS from node 25 to 729 and the rest has the same communication cost as the TDGS or lesser. The protocol manages to select a small number of replica to make a copy of a data. Communication costs are between 0.5 to 2.

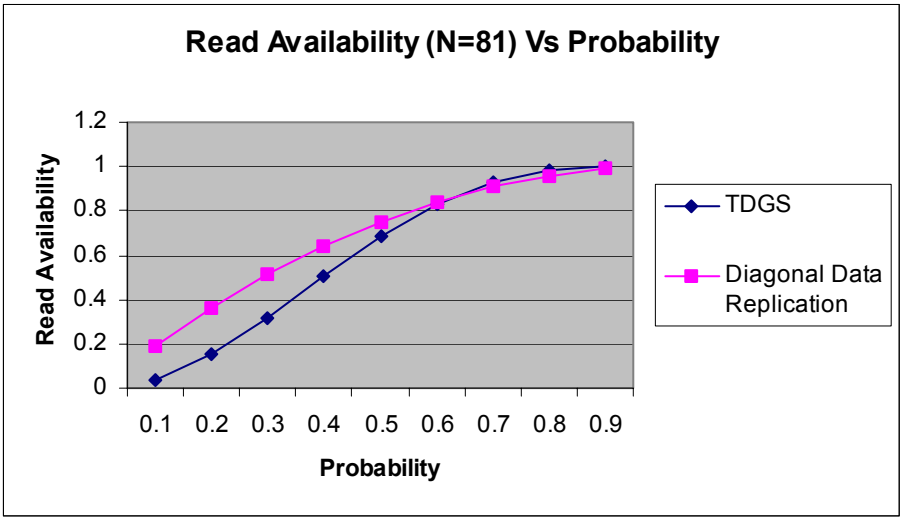


Fig. 7a. Read availability for 81 nodes

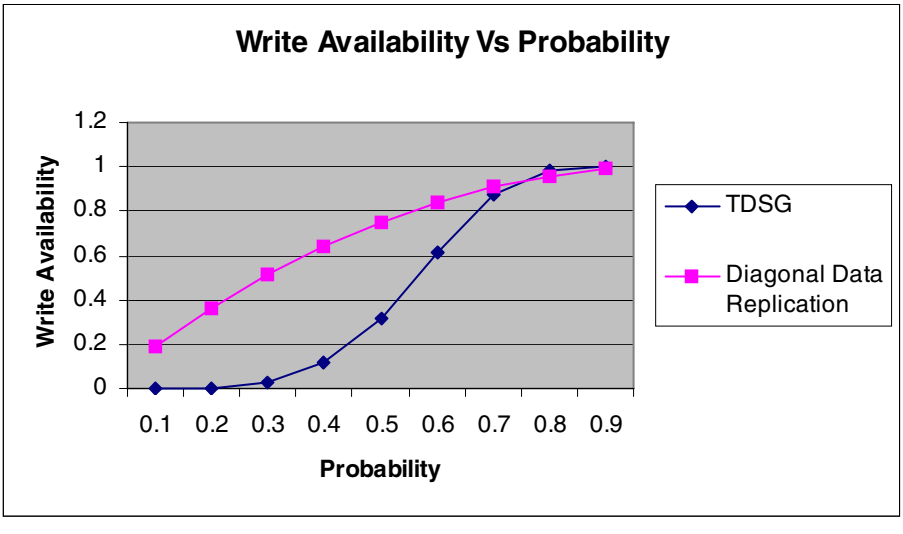


Fig. 7b. Write availability for 81 nodes

Fig. 6b shows the results for write operations for number of nodes from 25 to 729, where the communication cost for write in the diagonal replication protocol is lesser then TDGS because the number of selected replicas for our protocol is lower than TDGS. Our protocol is suitable for read and write operations.

From Fig. 6b, the results show that our protocol for write operation is lower in terms of communication cost. Our protocol needs only 1 to 3 selected replicas for write operation for number of nodes between 25 to 729. Fig. 7a shows the results of read availability for 81 nodes. It shows that the diagonal replication is high in both availability for read and write operations. Refer to Fig. 7b for results of write availability. The availability is high for both read and write because the replicas that are selected are in the middle location for all nodes in each quorum.

From Fig. 7a, the read availability for TDGS is higher 0.02 for probability 0.7 and 0.8 and the rest of the probability shows that Diagonal Data Replication protocol is higher. For write availability, TDGS is higher 0.02 at probability 0.8 and the rest shows that Diagonal Data Replication protocol is much higher.

6 Conclusions

In this paper, data replication protocol, put the selected replica in a diagonal way in a grid logical structured. The number of nodes in each quorum is odd so that the selected replica a chosen based on the middle replica in the diagonal nodes. This protocol improves the availability for both read and write operations in a low communication cost.

The simulator was created to study the performance of the diagonal replication protocol. The results demonstrate that the diagonal replication protocol can reduce both the read and write communication cost and availability compared to TDGS.

In the future, we will investigate the response time to access a range of data size in the grid environment and this investigation will be used to evaluate the performance of our diagonal replication protocol. Currently we have improved the protocol to a larger network and putting a certain number of nodes into tiers.

References

1. Krauter, K., Buyya, R., Maheswaran, M.: A Taxanomy and Survey of Grid Resource Management Systems for Distributed Computing. International Journal of Software Practice and Experience 32(2), 135–164 (2002)
2. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: Grid Services for Distributed System Integration. Computer 35(6), 37–46 (2002)
3. Ranganathan, K., Foster, I.: Identifying Dynamic Replication Strategies for a High Performance Data Grid. In: Proceedings of International Workshop on Grid Computing, Denver (November 2001)
4. Ranganathan, K., Foster, I.: Design and Evaluation of Replication Strategies for High Performance Data Grid. In: International Conference on Computing in High Energy and Nuclear Physics, Beijing (September 2001)
5. Ranganathan, K., Foster, I.: Grid Resource Management State of the Art and Future Trends. Academic Publisher (2003)

6. Lamehamedi, H., Szymanski, B., Shentu, Z., Deelman, E.: Data Replication Strategies in Grid Environment. In: Proceedings of ICAP'03, Beijing China, pp. 378–383. IEEE Computer Science Press, Los Alamitos (2002)
7. Lamehamedi, H., Shentu, Z., Szymanski, B.: Simulation of Dynamic Data Replication Strategies in Data Grids. In: Proceedings of the 17th International Symposium on Parallel and Distributed Processing, pp. 22–26 (2003)
8. Lamehamedi, H.: Decentralized Data Management Framework for Data Grids. Ph.D. Thesis, Rensselaer Polytechnic Institute Troy, New York (2005)
9. Tang, M., Lee, B., Yeo, C., Tang, X.: Dynamic Replication Algorithms for the Multi-tier Data Grid. Future Gener. Comput. Syst. 21, 775–790 (2005)
10. Mat Deris, M.: Efficent Access of Data Replication in Distibuted Database Systems. Ph.D. Thesis, Universiti Putra Malaysia, Malaysia (2001)
11. Mat Deris, M., Bakar, N., Rabiei, M., Suzuri, H.M.: Diagonal Replication on Grid for Efficient Access of Data in Distributed Database Systems. In: Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J.J. (eds.) ICCS 2004. LNCS, vol. 3038, pp. 379–387. Springer, Heidelberg (2004)
12. Mat Deris, M., Nathrah, B., Suzuri, M.H., Abu Osman, M.T.: Improving Data Availability Using Hybrid Replication Technique in Peer-to-Peer Environments. AINA 1, 593–598 (2004)
13. Mat Deris, M., Nathrah, B., Suzuri, M.H., Abu Osman, M.T.: Improving Data Availability Using Hybrid Replication Technique in Peer-to-Peer Environments. AINA 1, 593–598 (2004)

Efficient Shock-Capturing Numerical Schemes Using the Approach of Minimised Integrated Square Difference Error for Hyperbolic Conservation Laws

A.R. Appadu[1], M.Z. Dauhoo[2,⋆], and S.D.D.V. Rughooputh[3]

[1] Department of Mathematics, University Of Mauritius, Rep. of Mauritius
biprao2@yahoo.com
[2] Department of Mathematics, University Of Mauritius
m.dauhoo@uom.ac.mu
[3] Department of Physics, University Of Mauritius

Abstract. The relative phase error (RPE) is a measure of the amount of oscillations while the effective amplification factor measures the amount of damping in a numerical scheme. Recently [2], we have devised a new method baptised as the Minimised Integrated Square Difference Error, **(MISDE)** for obtaining efficient shock-capturing schemes. In this work, we show based on our concept that the 4-6 Low-Dissipation and Dispersion Runge-Kutta (4-6 LDDRK) [6] is very efficient to capture shocks. Using **MISDE**, we obtain a modified form of the Fromm's scheme in 1-D which is less dispersive. An optimal value of the parameter, ϵ at $cfl = 0.9$ in the $LW - \epsilon$ [10] scheme is also computed. It is shown that the S_α^β scheme [5] is more efficient than the 2-D MacCormack scheme to capture shocks. We solve the 2-D circular explosion problem [11] using the LWLF6 and MCLF6 [1] schemes at some cfl numbers and show that shock capturing ability is dependent on the cfl number.

Keywords: 4-6 Low-Dissipation and Dispersion Runge Kutta, $LW - \epsilon$, S_α^β, Minimised Integrated Square Difference Error.

Nomenclature: $I = \sqrt{(-1)}$; k: time step; h: spatial step (in 1-D); PDE: Partial differential equation; β: advection velocity; r: $cfl/Courant$ number; $r = \dfrac{\beta k}{h}$; θ: wave number (in 1-D); i, j: mesh point locations in x and y directions; w: phase angle in 1-D; $w = \theta h$; n: time level; f, g: flux functions; ϕ_x: phase angle in x-direction; ϕ_y: phase angle in y-direction; ξ: amplification factor; A, B: Jacobian matrices; $\lambda = A\ r_x$; $\mu = B\ r_y$; $r_x = \dfrac{k}{\Delta x}$; $r_y = \dfrac{k}{\Delta y}$; RPE: relative phase error per unit time step; AF: amplification factor; $AFM = |AF|$; EAF: effective amplification factor per unit time step; **ISDE**: Integrated Square Difference Error; **MISDE**: Minimised Integrated Square Difference Error.

⋆ Corresponding author.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 774–789, 2007.
© Springer-Verlag Berlin Heidelberg 2007

1 Introduction

The concept of **MISDE** has been inspired by the work of Liska et al [8]. The Lax-Wendroff (LW) and the 2-step Lax Friedrichs (LF) schemes have mutual opposite drawbacks in their dissipation and dispersion effects [7]. The LW scheme is essentially dispersive and generates oscillation close to regions of steep gradients. On the other hand, the LF scheme is non-oscillatory but dissipative in regions of steep gradients. Composition of several steps of the dispersive scheme followed by one step of the LF scheme results in adequate accuracy by application of two numerical schemes which have counter remainder effects [7]. The composite effects of dissipation and dispersion are well worth noticing [9]. Rational adaptation of the composite effects, especially the control of dispersion and dissipation is really a difficult and an important problem [9]. When dissipation dominates dispersion, parasitic oscillations of dispersion are restrained [9]. In numerical schemes, dissipation can be quantified in terms of the EAF while dispersion is linked with the RPE.

In a previous work, [1], we obtained the cfl number whereby dissipation curbs dispersion in a numerical scheme. We termed such a scheme as efficient shock capturing. The concept of **MISDE** has been devised in [2]. In this technique, the square of the difference between the RPE and the EAF is minimised with respect to cfl number. We showed in [2] that this new concept can be applied to some schemes in 1-D namely MacCormack (MC), 2-step Lax-Friedrichs [8] (LF), classic Leap-Frog (LFR) and the composite MacCormack/Lax-Friedrichs scheme (MCLFn) with n = 2, 3, 4 and 5. The optimal cfl number for these schemes have been obtained as 1.0 using **MISDE**. Using the shift condition [11], we observe that the best cfl number for the MC and LF schemes is 1.0. In order to verify and validate the new concept in [2], we have solved the 1-D shallow water problem and the following problems based on wave propagation: Gaussian problem, Boxcar function and sine function in 1-D. We have also used the concept in [2] to show that the following scheme in 1-D:

$$u_i^{n+1} = \delta u_i^n + (1-\delta)\tilde{u}_i \tag{1}$$

where

$$\bar{u}_i = u_i^n - r(u_{i+1}^n - u_i^n) \tag{2}$$

and

$$\tilde{u_i} = \bar{u}_i - r(\bar{u}_i - \bar{u}_{i-1}) \tag{3}$$

is most efficient to capture shocks when $\delta = \dfrac{1}{2}$, i.e when the scheme is the 2-step MacCormack scheme. Recently, in [3], the concept of **MISDE** has been used to explain why the 2-step Lax-Friedrichs scheme captures shocks better than the usual Lax-Friedrichs scheme (ULF) and does not exhibit odd/even decoupling which afflicts the usual Lax-Friedrichs scheme.

Compared to the method given in [1] for locating the optimal cfl, **MISDE** is comparatively easier in the sense that in [1], we have to locate the optimal cfl using a surface plot whereas in **MISDE**, we simply have a line graph.

The paper is organised as follows. In section 2, we describe some schemes which satisfy the shift condition. We then use **MISDE** to compare the shock capturing ability of the 4-6 LDDRK and RK4 schemes. We then obtain an optimal value of ϵ in the $LW-\epsilon$ scheme at $cfl=0.9$ and also obtain the optimal cfl number at a given value of ϵ, for instance 0.1 [10]. In both of these cases, the results are validated using the 1-D shallow water problem. In section 5, we seek for the optimal cfl for the 1-D Fromm's scheme and a modified Fromm's scheme is derived in section 6. We use **MISDE** to obtain a modified Fromm's scheme with lower dispersion than the classical Fromm's scheme, in general for $0 \leq r \leq 1$ and $w \in (0, 1.0)$, where r is the cfl number. The latter scheme is baptised as 'Fromm 42-58' which uses Beam-Warming (BW) and Lax-Wendroff (LW) in the ratio 42:58. In section 7, **MISDE** has been applied to 2-D numerical schemes. It is shown that the S_α^β scheme [5] is more efficient as compared to the 2-D MacCormack scheme in order to capture shocks. Lastly, we obtain the optimal cfl number for LWLFn and MCLFn schemes in 2-D for $n=2,3,4,5$ and 6 and solve the 2-D circular explosion problem [11] using LWLF6 and MCLF6. In section 9, we conclude the main results of this work.

2 Shift Condition

The upstream (first order upwind) differencing method has been shown to satisfy the shift condition in [12] (Kutler and Lomax, 1971). When the upstream scheme is used to discretize the the 1-D linear advection equation,

$$u_t + \beta\ u_x = 0, \tag{4}$$

we get

$$\frac{u_i^{n+1}-u_i^n}{k} + \beta\ \frac{u_i^n - u_{i-1}^n}{h} = 0. \tag{5}$$

If $r=1$, the scheme reduces to $u_i^{n+1} = u_{i-1}^n$, which is equivalent to solving the wave equation using the method of characteristics. Thus, the ideal cfl number for the upstream scheme is 1. We now consider a scheme which does not satisfy the shift condition. The fourth-order in space, second order in time, space centred Leap-Frog scheme, (42LFR) [4] is given by:

$$u_i^{n+1} = u_i^{n-1} + \frac{r}{6}\ \left(u_{i+2}^n - u_{i-2}^n\right) - \frac{4}{3}r\ \left(u_{i+1}^n - u_{i-1}^n\right), \tag{6}$$

and is stable for $0 \leq r \leq 1$.

Using **MISDE**, we obtain the optimal cfl number as 0.18. Numerical experiments performed in [14] indicate that the solution obtained is more accurate for $r < 0.5$ where cancellation between time and space truncation errors occur. It has been pointed out in [14] that for most schemes, the most accurate results are

obtained for cfl close to the stability limit but such is not the case with 42LFR scheme.

We conclude that though a scheme is optimally stable, it may not satisfy the shift condition at $r = 1$ and that the optimal cfl number need not be 1.0, for efficient shock capturing as shown by numerical experiments in [14] and which we have proved using **MISDE**.

3 Runge Kutta Schemes

In this section, we compare two multistage Runge-Kutta time marching methods for MacCormack-type schemes [6]. A general six-stage Runge-Kutta MacCormack-type scheme can be defined as:

$$h^{(1)} = -k\ \delta^F[F(U^n)], \tag{7}$$

$$h^{(2)} = -k\ \delta^B[F(U^n + \alpha_2\ h^{(1)})], \tag{8}$$

$$h^{(3)} = -k\ \delta^F[F(U^n + \alpha_3\ h^{(2)})], \tag{9}$$

$$h^{(4)} = -k\ \delta^B[F(U^n + \alpha_4\ h^{(3)})], \tag{10}$$

$$h^{(5)} = -k\ \delta^F[F(U^n + \alpha_5\ h^{(4)})], \tag{11}$$

$$h^{(6)} = -k\ \delta^B[F(U^n + \alpha_6\ h^{(5)})], \tag{12}$$

and

$$U^{n+1} = U^n + \beta_1\ h^{(1)} + \beta_2\ h^{(2)} + \beta_3\ h^{(3)} + \beta_4\ h^{(4)} + \beta_5\ h^{(5)} + \beta_6\ h^{(6)}, \tag{13}$$

where δ^F: forward spatial difference and δ^B: backward spatial difference.

The coefficients for the 4-6 LDDRK and RK4 schemes are given in [6]. These 2 schemes are fourth-order accurate in a linear sense. However, the 4-6 LDDRK reduces to third-order accuracy for non-linear problems. The latter uses a two-step marching cycle; first step has four stages and the second step has six stages.

Here, we investigate about an optimal cfl number for the 4-6 LDDRK and RK4 schemes [6].

We conduct a stability analysis of the 4-6 LDDRK and RK4 using Von Neumann method. Fig.(1) shows the plots of AFM v/s the cfl number v/s w for these 2 numerical schemes.

It is seen that the LDDRK scheme is stable for $0 \leq r \leq 1.56$ while the RK4 scheme is stable for $0 \leq r \leq 1.72$. Using **MISDE**, the optimal cfl number for 4-6 LDDRK and RK4 are respectively 1.0 and 0.35. We observe that the quantity, **ISDE** is in general smaller for LDDRK 4-6 as compared to RK4. Also, **ISDE**

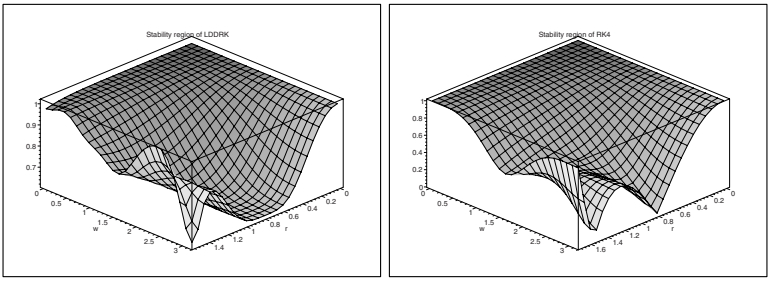


Fig. 1. Plot of AFM v/s r v/s w to obtain stability regions of 4-6 LDDRK and RK4 schemes

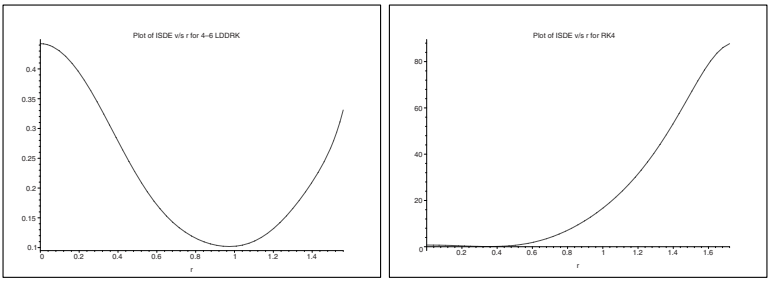


Fig. 2. Variation of **ISDE** v/s cfl number for 4-6 LDDRK and RK4 schemes

is large in RK4 for high cfl numbers. These phenomena are depicted in Fig.(2). The optimal cfl numbers obtained confirm that not in all cases the optimal cfl number is close to the stability limit, as pointed out in [14].

4 $LW - \epsilon$ Scheme

Liu Ruxun et al. [10] designed a stable, high resolution scheme by controlling the numerical effects of dissipation and dispersion. They used a simple approach by reforming the LW scheme by parallel replacement upwards in an attempt to construct a monotonic and positive scheme.

A scheme consistent with Eq.(4) can be written as [10]:

$$u_i^{n+1} = a_1\ u_{i+1}^n + a_0\ u_i^n + a_{-1}\ u_{i-1}^n. \tag{14}$$

The $LW - \epsilon$ scheme is given by

$$u_i^{n+1} = \frac{1}{2}(r^2 - r + \epsilon)\ u_{i+1}^n + (1 - r^2 - \epsilon)\ u_i^n + \frac{1}{2}(r^2 + r + \epsilon)\ u_{i-1}^n. \tag{15}$$

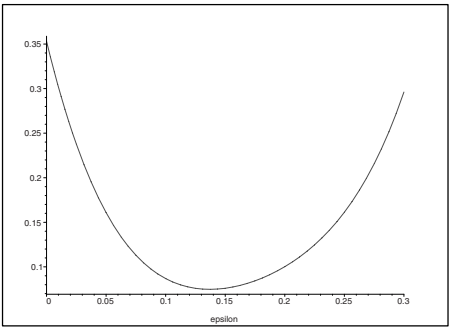


Fig. 3. Plot of **ISDE** v/s ϵ at $cfl = 0.9$ to obtain the optimal value of ϵ in the $LW - \epsilon$ for efficient shock capturing

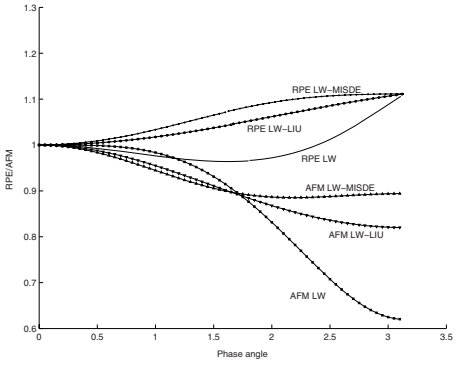


Fig. 4. Plot of RPE and AFM at $cfl = 0.9$ for LW and $LW - \epsilon$ schemes; LW-MISDE denotes $LW - \epsilon$ with $\epsilon = 0.13706$ and LW-LIU denotes $LW - \epsilon$ with $\epsilon = 0.1$

By fixing $h = 0.1$, $r = 0.9$, it was found that choosing $\epsilon = 0.1$ results in the following coefficients: $a_1 = 0.005$, $a_0 = 0.09$, $a_{-1} = 0.905$. This renders the $LW - \epsilon$ still second order accurate. The numerical results obtained using the $LW - \epsilon$ scheme are satisfactory as shown on page 531, Fig.(3) in [10].

We use **MISDE** to seek for an optimal value of ϵ at $cfl = 0.9$ i.e the value of ϵ number at which the $LW - \epsilon$ captures sharp shocks efficiently.

The amplification factor of $LW - \epsilon$ scheme at $cfl = 0.9$ is obtained as:

$$\xi = 0.19 - \epsilon + (\epsilon + 0.81)\ \cos(w) - I\ (0.9\ \sin(w)). \tag{16}$$

Fig.(3) shows a plot of the **ISDE** v/s ϵ for the $LW - \epsilon$ scheme at $cfl = 0.9$. From the graph, we deduce that the optimal value of ϵ is 0.13706. A comparison of the RPE and AFM of the $LW - \epsilon$ scheme for $\epsilon = 0.1$ and $\epsilon = 0.13706$ and of the

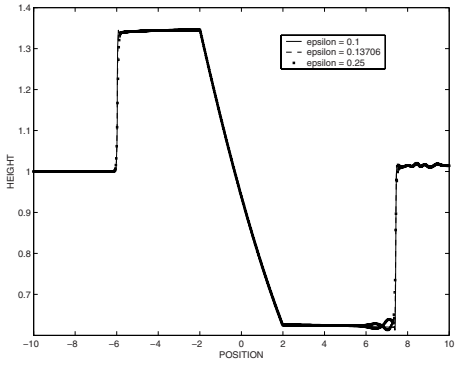


Fig. 5. Comparison of heights for the 1-D shallow water problem at $cfl = 0.9$ using 3 different values of ϵ

Lax-Wendroff scheme is made in Fig.(4). It is observed that these 2 versions of $LW - \epsilon$ have almost the same RPE. However, when $\epsilon = 0.1$ we have in general more dissipation as compared to $\epsilon = 0.13706$. This shows that **MISDE** can be effectively used to find the optimal parameter, ϵ at a particular cfl number for efficient shock capturing.

In general when a numerical scheme is TVD, the amount of oscillations is bounded. We now test for monotone and Total Variation Diminishing (TVD) properties [13] for 3 different values of ϵ: 0.1, 0.13706 and 0.25, all at $cfl = 0.9$. The $LW - \epsilon$ scheme at $cfl = 0.9$ using $\epsilon = 0.1, 0.13706$ and 0.25 reduce to the following schemes:

$$u_i^{n+1} = 0.005\ u_{i-1}^n + 0.09\ u_i^n + 0.905\ u_{i+1}^n, \tag{17}$$

$$u_i^{n+1} = 0.002353\ u_{i-1}^n + 0.05294\ u_i^n + 0.92353\ u_{i+1}^n, \tag{18}$$

and

$$u_i^{n+1} = 0.08\ u_{i-1}^n - 0.06\ u_i^n + 0.98\ u_{i+1}^n, \tag{19}$$

respectively.

We deduce that the $LW - \epsilon$ at $cfl = 0.9$ is not monotone when $\epsilon = 0.25$ but is monotone and hence TVD when $\epsilon = 0.1$ and $\epsilon = 0.13706$. We now solve the 1-D shallow water problem [8] at $cfl = 0.9$ using 3 different values of ϵ: 0.1, 0.13706 and 0.25. The results are shown in Fig.(5).

Out of these 3 different values of ϵ, $\epsilon = 0.13706$ gives the best results with least amount of oscillations. We observe that when using $\epsilon = 0.1$, there is a slight overshoot just after the first shock. There is a larger undershoot just after the second shock when $\epsilon = 0.1$ as compared to $\epsilon = 0.13706$. Considerable computational noise manifested as pre-shock and post-shock oscillations occur near the second discontinuity with $\epsilon = 0.25$.

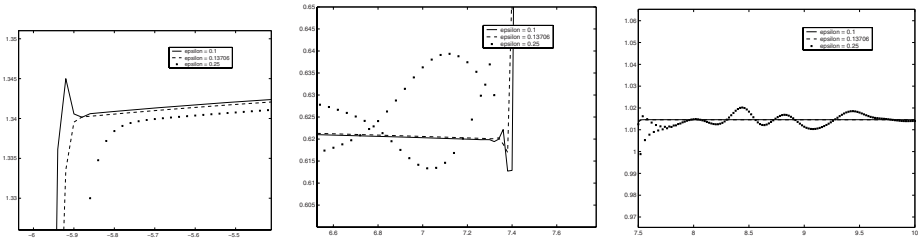


Fig. 6. Zoomed regions close to discontinuities for Fig.(5)

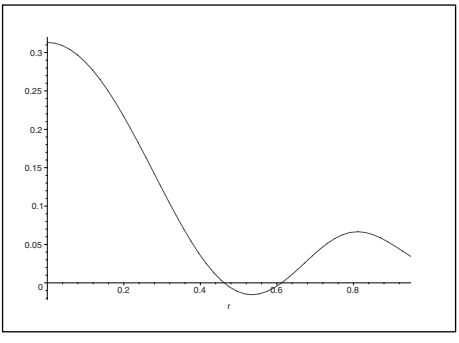


Fig. 7. Plot of **ISDE** v/s r with $\epsilon = 0.1$ to obtain the optimal value of r in the $LW - \epsilon$ for efficient shock capturing

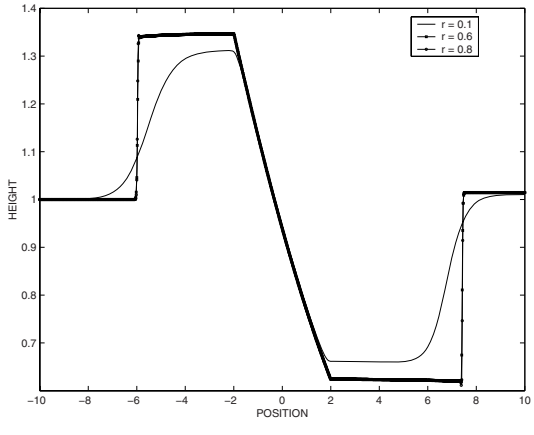


Fig. 8. Comparison of heights for the 1-D shallow water problem at 3 different cfl numbers with $\epsilon = 0.1$, in $LW - \epsilon$ scheme

We next fix the value of ϵ and use **MISDE** to obtain the corresponding optimal cfl number for efficient shock capturing. A plot of the **ISDE** v/s r is illustrated in

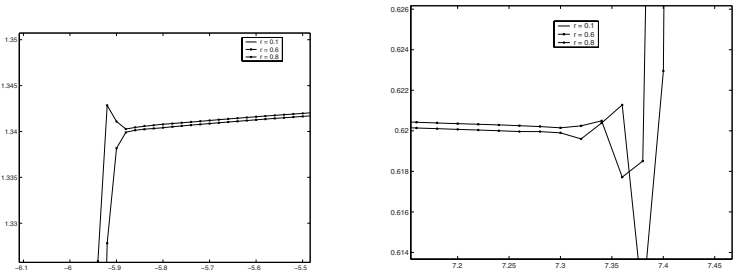


Fig. 9. Zoomed regions close to discontinuities for Fig.(8)

Fig.(7) which depicts that when $\epsilon = 0.1$, the optimal cfl is approximately 0.4 or 0.6. Figs.(8) and (9) illustrate the results of the 1-D shallow water flow at some cfl numbers with $\epsilon = 0.1$. At $r = 0.1$, the results are highly dissipative and distorted. The results are more dispersive at $cfl = 0.8$ as compared to $cfl = 0.6$.

The results which are illustrated in Figs.(5), (6), (8) and (9) show that intrinsic characteristics of the schemes mainly depend on the coefficients of the numerical scheme which affect the amount of dissipation and dispersion in the numerical scheme.

5 Fromm's Scheme in 1-D

The Fromm's scheme of zero-average phase error in 1-D uses an average of Lax-Wendroff and the Beam Warming (BW) scheme. The BW [12] method has a predominantly leading phase error for $0 < r < 1$ and a predominantly lagging phase error for $1 < r < 2$. A single expression for the Fromm's scheme in 1-D is given by:

$$u_i^{n+1} = \frac{r}{4}(r-1)\, u_{i-2}^n + \frac{r}{4}(5-r)\, u_{i-1}^n + \Big(1 - \frac{3}{4}r - \frac{r2}{4}\Big)\, u_i^n + \frac{1}{4}(r^2-r)\, u_{i+1}^n. \quad (20)$$

The maximum stability limit is obtained as $r = 1$. Using $r = 1$ in Eq.(20), the equation reduces to $u_i^{n+1} = u_{i-1}^n$. Hence, shift condition is satisfied when $r = 1$. Using **MISDE**, we have obtained the optimal cfl as 1.0.

6 Modified Fromm's Scheme: 1-D Fromm 42-58

In this section, we investigate whether the ratio of 1 : 1 of LW to BW scheme in the 1-D Fromm's zero-average phase error scheme is optimal. We consider the Fromm's scheme in the following form:

$$u_i^{n+1} = \lambda\Big(c - r(c-b) + \frac{1}{2}r(r-1)(c-2b+a)\Big) + (1-\lambda)\Big(c - \frac{r}{2}(d-b) + \frac{r^2}{2}(d-2c+b)\Big), \quad (21)$$

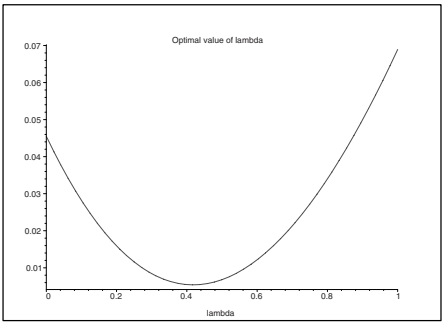


Fig. 10. Plot of **ISDE** v/s r to obtain the optimal value of λ in the modified Fromm's scheme for efficient shock capturing

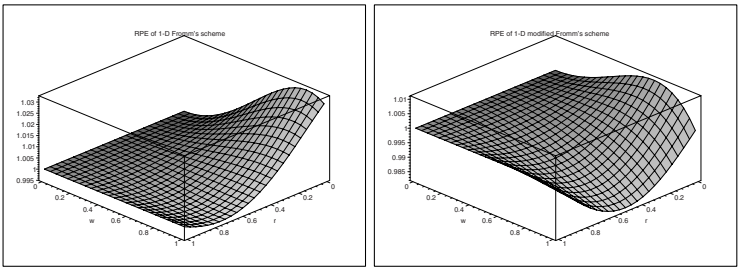


Fig. 11. Comparison of RPE for 1-D Fromm zero average phase error and "1-D Fromm 42-58" schemes

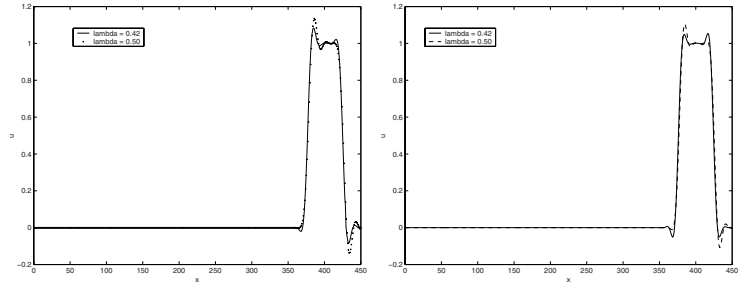


Fig. 12. Comparison of results for propagation of boxcar at different cfl numbers using 1-D Fromm zero average phase error and '1-D Fromm 42-58' schemes

where $a = u^n_{i-2}$, $b = u^n_{i-1}$, $c = u^n_i$, $d = u^n_{i+1}$ and $\lambda \in [0, 1]$. The quantity, **ISDE** for the scheme given by Eq.(21) is obtained in terms of λ as

$$ISDE_{fromm} = 0.0455 - 0.19912\lambda + 0.26432\lambda^2 - 0.04178\ \lambda^3. \qquad (22)$$

The optimal value of λ is 0.42 as shown in Fig.(10). Fig.(11) shows the 3-D plots of RPE v/s r v/s w for 1-D Fromm zero average phase error and '1-D Fromm

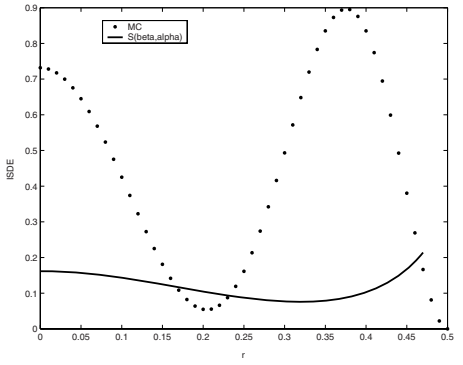


Fig. 13. Comparison of the variation of **ISDE** with respect to cfl number for 2-D MacCormack and S_α^β schemes

42-48' schemes. It is seen that the RPE varies from 0.995 to 1.03 in the case of the 1-D Fromm's scheme but in the case of '1-D Fromm 42-58', the RPE lies over a smaller range: from 0.985 to 1.01 for $w \in [0, 1]$ and $r \in [0, 1]$.

Fig.(12) illustrates the results of the propagation of the Boxcar function using these two schemes at cfl numbers 0.1 and 0.25 respectively.

7 S_α^β and 2-D MacCormack Schemes

Using **MISDE**, we now prove that the 2-D optimal S_α^β scheme [5] has better shock capturing ability than the 2-D MacCormack.

The amplification factor of the 2-D MacCormack for the special case $\phi_x = \phi_y = w$ for the 2-D linear advection equation, described by $U_t = f_x(U) + g_y(U)$, with $f = AU$ and $g = BU$, is given by:

$$\xi_{M_2} = 1 - 8r^2 \ \sin^2\left(\frac{w}{2}\right) - I[2r \ \sin(w)]. \tag{23}$$

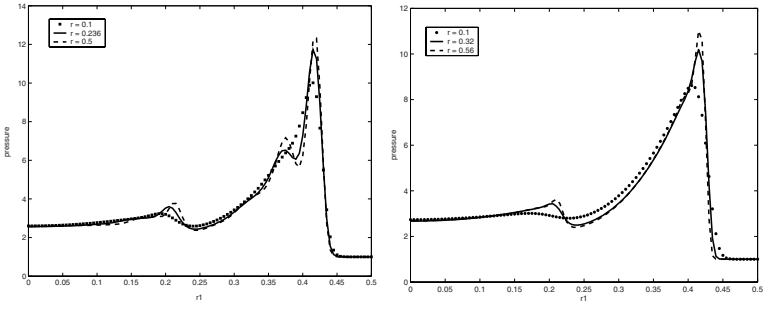


Fig. 14. Distribution of pressure along the line $y = 0.5$ where $r1 = x - 0.5 \geq 0$ using LWLF6 and MCLF6 schemes at different cfl numbers

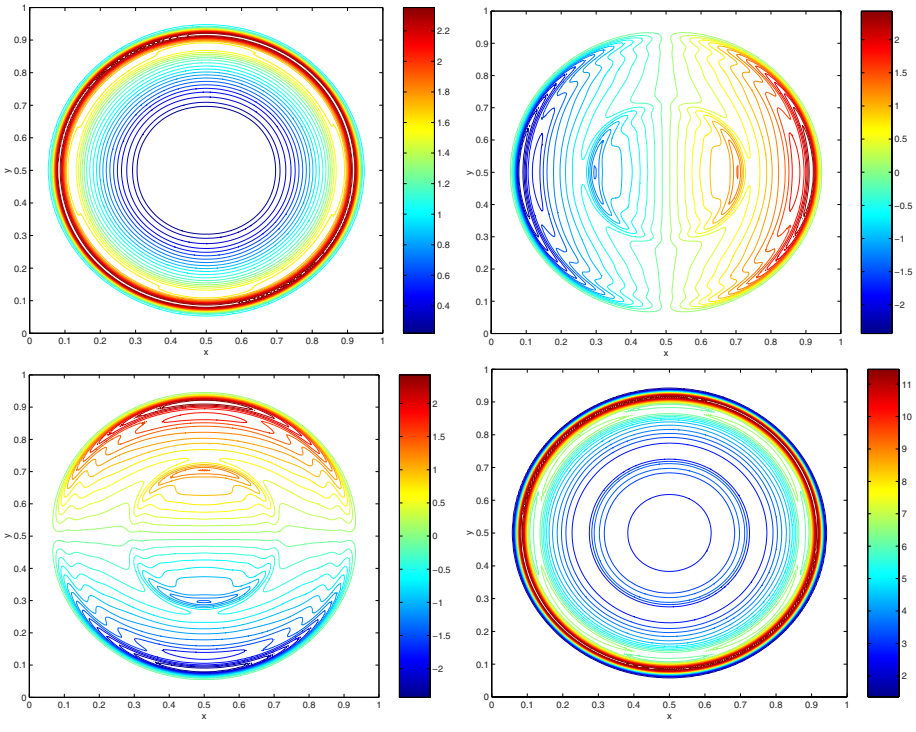


Fig. 15. Cylindrical explosion, isolines of the solution obtained by LWLF6 scheme with optimal cfl number 0.236 at time 0.0275 and mesh 200×200, the plots show density (top left), horizontal velocity (top right), vertical velocity (bottom left) and pressure (bottom right).

The corresponding expression for the amplification factor of the S_α^β is computed as:

$$\xi_{S(\alpha,\beta)} = 1 + r^2(2\ \cos(w) + \ \cos^2 w - 3) - I[2\ r\ \sin(w)]. \tag{24}$$

Two points to be noted are:

(i) In the case of S_α^β, the values of **ISDE** lie in the range 0.08 to 0.232 for $0 \leq r \leq 0.47$ while in the case of 2-D MacCormack, the corresponding values vary from 0 to 1.0 as shown in Fig.(13). Thus, the values of the **ISDE** at a given cfl number are in general closer to zero for the S_α^β as compared to the 2-D MacCormack. Therefore, we would expect S_α^β to capture sharper shocks with less dispersion. This is confirmed by numerical experiments for a transonic airfoil computation in [5].
(ii) For the 2-D MacCormack scheme, the optimal cfl number is 0.5 but in the case of S_α^β, the optimal cfl number is 0.32.

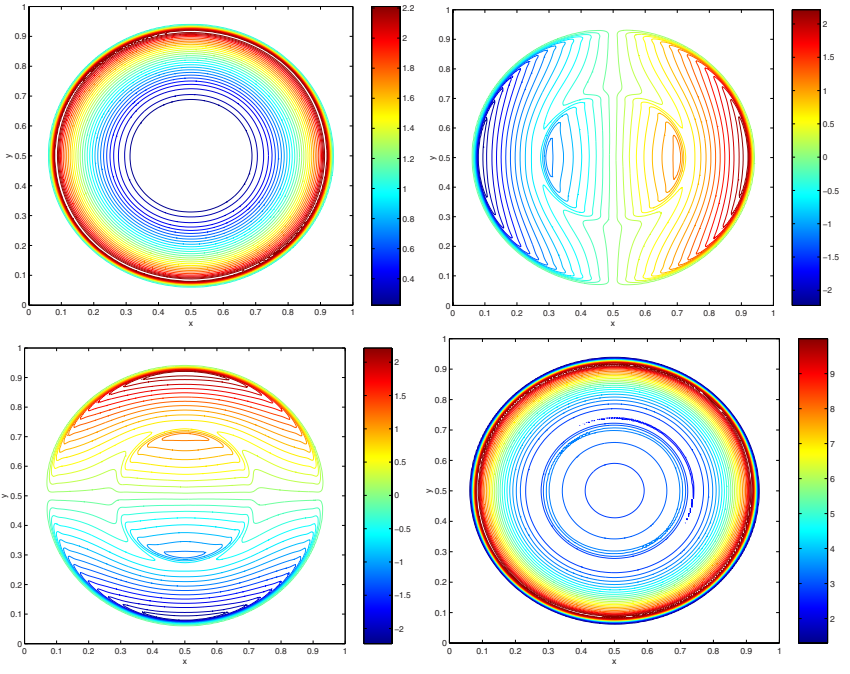


Fig. 16. Cylindrical explosion, isolines of the solution obtained by MCLF6 scheme with optimal cfl number 0.32 at time 0.05 and mesh 200×200, the plots show density (top left), horizontal velocity (top right), vertical velocity (bottom left) and pressure (bottom right)

8 Application of MISDE in 2-D Composite Schemes

Expressions for the Integrated Square Difference Error for the LWLF2, LWLF3, LWLF4, LWLF5 and LWLF6 schemes in 2-D are given respectively by:

$$\int_0^{1.55} (EAF - RPE)^2 \, dw = 1.9835 \times 10^{-1} - 8.5628r^2 + 105.7r^4 - 263.17r^6 + 186.81r^8 + 3.8716 \times 10^{-1} r^{10} + 7.446 \times 10^{-2} r^{12} + 1.0527 \times 10^{-4} r^{14} + 1.0096 \times 10^{-5} r^{16}, \quad (26)$$

$$\int_0^{1.45} (EAF - RPE)^2 \, dw = 2.1694 \times 10^{-1} - 8.8011r^2 + 102.3r^4 - 252.36r^6 + 179.69r^8 + 2.1857 \times 10^{-1} r^{10} + 3.4728 \times 10^{-2} r^{12} + 2.7476 \times 10^{-5} r^{14} + 2.1924 \times 10^{-6} r^{16}, \quad (27)$$

$$\int_0^{1.4} (EAF - RPE)^2 \, dw = 2.2798 \times 10^{-1} - 8.9449r^2 + 100.7r^4 - 245.73r^6 + 174.17r^8 + 1.4814 \times 10^{-1} r^{10} + 2.1374 \times 10^{-2} r^{12} + 1.1595 \times 10^{-5} r^{14} + 8.4241 \times 10^{-7} r^{16} \quad (28)$$

and

$$\int_0^{1.35} (EAF - RPE)^2 \, dw = 2.1799 \times 10^{-1} - 8.4275r^2 + 93.42r^4 - 225.32r^6 + 158.4r^8 + 1.0008 \times 10^{-1} r^{10} + 1.3102 \times 10^{-2} r^{12} + 5.1863 \times 10^{-6} r^{14} + 3.4261 \times 10^{-7} r^{16}. \quad (29)$$

The corresponding expressions for MCLF2, MCLF3, MCLF4, MCLF5 and MCLF6 schemes in 2-D are

$$\int_0^{\pi} (EAF - RPE)^2 \, dw = 2.6387 \times 10^{-2} - 1.5316 \times 10^{-1} r^2 + 35.908 r^4 - 177.22 r^6 + 220.15 r^8 - 2.1056 \times 10^{-1} r^{10} + 5.6863 r^{12} + 2.0565 \times 10^{-2} r^{14} + 3.856 \times 10^{-2} r^{16}, \quad (30)$$

$$\int_0^{\pi} (EAF - RPE)^2 \, dw = 2.0517 \times 10^{-1} - 7.222 r^2 + 84.353 r^4 - 323.64 r^6 + 391.24 r^8 - 2.4912 \times 10^{-1} r^{10} + 5.047 r^{12} + 9.14 \times 10^{-3} r^{14} + 1.7137 \times 10^{-2} r^{16}, \quad (31)$$

$$\int_0^{3.0} (EAF - RPE)^2 \, dw = 3.3254 \times 10^{-1} - 9.6408 r^2 + 94.01 r^4 - 342.12 r^6 + 414.02 r^8 + 1.3601 \times 10^{-1} r^{10} + 3.4152 r^{12} + 5.1123 \times 10^{-3} r^{14} + 7.4567 \times 10^{-3} r^{16}, \quad (32)$$

$$\int_0^{2.9} (EAF - RPE)^2 \, dw = 3.9553 \times 10^{-1} - 10.47 r^2 + 94.894 r^4 - 336.79 r^6 + 407.88 r^8 + 2.9954 \times 10^{-1} r^{10} + 2.4295 r^{12} + 3.2022 \times 10^{-3} r^{14} + 3.8445 \times 10^{-3} r^{16}, \quad (33)$$

and

$$\int_0^{2.8} (EAF - RPE)^2 \, dw = 4.051 \times 10^{-1} - 10.195 r^2 + 88.935 r^4 - 311.28 r^6 + 377.28 r^8 + 3.885 \times 10^{-1} r^{10} + 1.7178 r^{12} + 2.1225 \times 10^{-3} r^{14} + 2.0849 \times 10^{-3} r^{16}, \quad (34)$$

respectively.

On minimising the Integrated Square Difference Error, we get the optimal cfl number for each of the 6 variants of LWLFn and MCLFn schemes.

The optimal cfl numbers for LWLF2, LWLF3, LWLF4, LWLF5 and LWLF6 are equal to 0.20, 0.22, 0.23, 0.234 and 0.236 respectively. In the case of MCLF3, MCLF4, MCLF5 and MCLF6, the optimal cfl numbers are 0.25, 0.29, 0.3 and 0.32 respectively. A cfl close to zero is optimal for MCLF2 scheme.

8.1 Cylindrical Explosion for 2D Euler Equation

Our aim is to compute a cylindrical explosion problem of two-dimensional Euler equations [11]. The computational domain comprises of a unit square. There is a high pressure region around the centre with radius 0.1. The initial density over the entire domain is 1 while the pressure inside and outside the central region are 100 and 1, respectively.

The results obtained with LWLF6 scheme for the explosion problem are dependent on the cfl number chosen. At a cfl of 0.1, there is considerable damping. A cfl of 0.5 induces oscillations in 2 regions, close to $r1 = 0.22$ and $r1 = 0.4$. For the MCLF6 scheme, a cfl of 0.32 gives better results with minimum damping and oscillations as compared to cfl numbers 0.1 and 0.56. The results are shown in Figs.(14), (15) and (16).

9 Conclusion

In this work, we obtained an optimal cfl number for some schemes in 1-D namely, 42LFR, 4-6 LDDRK and RK4. **MISDE** is used to show that in general the 4-

6 LDDRK has better shock capturing ability than RK4 schemes in 1-D and also that the S_{α}^{β} scheme is more efficient than the 2-D MacCormack scheme to resolve discontinuities. We have obtained an optimal value of the parameter, ϵ at $cfl = 0.9$ and also the optimal cfl number at $\epsilon = 0.1$ for the $LW - \epsilon$ scheme. A modified form of the Fromm's scheme which is in general slightly less dispersive than the 1-D Fromm's zero-average phase error scheme has been obtained. We have also computed the optimal cfl number for some variants of the two composite schemes in 2-D namely LWLFn and MCLFn, and solve a 2-D cylindrical explosion problem with LWLF6 and MCLF6 schemes. From the numerical results obtained in 1-D and 2-D, we observe that the concept of **MISDE** is a versatile and powerful concept which can be used in order to adjust certain parameters such as the cfl number in order to obtain results with efficient shock capturing ability.

Acknowledgements

Mr A.R.APPADU is an MPhil/Phd student at the University of Mauritius and has been partially supported by the Tertiary Education Commission (TEC) for the research through a bursary.

References

1. Appadu, A.R., Dauhoo, M.Z., Rughooputh, S.D.D.V.: Analysis of Two Composite Numerical Schemes Based on Lax-Wendroff and MacCormack schemes with Lax-Friedrichs Scheme. In: SIAM Conference on Mathematical and Computational Issues in the Geosciences. Palais des Papes, the International Conference Center Avignon, FRANCE, June 7-10, 2005,
2. Appadu, A.R., Dauhoo, M.Z., Rughooputh, S.D.D.V.: Optimisation of numerical schemes using the Minimised Integrated Square Difference Error. Presented at Research Week 2007 held from 16-20 January 2007 at the University of Mauritius, Mauritius (To be included in the book of abstracts)
3. Appadu, A.R., Dauhoo, M.Z.: A Note on the Two Versions of Lax-Friedrichs Scheme. In: Proceedings of the 2007 World Congress in Computer Science, Computer Engineering, and Applied Computing- Scientific Computing, Monte Carlo Resort, Las Vegas, Nevada, USA, 25-28 June 2007, CSREA Press (2007)
4. Babarsky, J.B., Sharpley, R.C.: Expanded Stability Through Higher Temporal Accuracy for Time-Centred Advection Schemes. Monthly Weather Review 125, 1277–1295 (1997)
5. Hirsch, C.: Numerical Computation of Internal and External flows, vol. 2. John Wiley and Sons, Chichester (1990)
6. Hixon, R., Turkel, E.: High-Accuracy Compact MacCormack-Type Schemes for Computational Aeroacoustics. National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191, Contractor Report (October 1998)
7. Jiwen, W., Ruxun, L.: The composite finite volume method on unstructured meshes for the two-dimensional shallow water equations. International Journal for Numerical Methods in Fluids 37, 933–949 (2001)

8. Liska, R., Wendroff, B.: Composite Schemes for conservation laws. SIAM J.Numer. Anal. 35, 2250–2271 (1998)
9. Ru-xun, L.: The Remainder-Effect Analysis Of Finite Difference Schemes and The Applications. Applied Mathematics and Mechanics 16, 87–96 (1995)
10. Ruxun, L., Mengping, Z., Ji, W., Xiao-Yuan, L.: The designing approach of difference schemes by controlling the remainder-effect. International Journal for Numerical Methods in Fluids 31, 523–533 (1999)
11. Tang, H., Warnecke, G.: A Class of High Resolution Schemes for Hyperbolic Conservation Laws and Convection-Diffusion Equations with Varying Time and Space Grids. Preprint-samtliche Rechte verbleiben den Autoren (2003)
12. Tannehill, J.C., Anderson, D.A., Pletcher, R.H.: Computational Fluid Mechanics and Heat Transfer. 2nd edn., Taylor and Francis, series in computational and physical processes in mechanics and thermal sciences
13. Toro: Non-Oscillatory Advection Schemes for Model Equations. Lecture Notes
14. http://twister.ou.edu/CFD2005/Chapter3.pdf

Improvement on Real-Time Face Recognition Algorithm Using Representation of Face and Priority Order Matching

Tae Eun Kim[1], Chin Hyun Chung[1], and Jin Ok Kim[2]

[1] Department of Information and Control Engineering,Kwangwoon University, 447-1, Wolgye-dong, Nowon-gu, Seoul, 139-701, Korea
chung@kw.ac.kr
[2] Faculty of Multimedia, Daegu Haany University, 290, Yugok-dong,Gyeongsan-si, Gyeongsangbuk-do, 712-715, Korea
bit@dhu.ac.kr

Abstract. Face recognition is an important part of today's emerging biometrics and application of security. In this paper we will be proposing an algorithm for Real-Time face recognition algorithm using representation of face and priority order matching. It is common to register a lot of faces into a database for to improve face recognition rate. We used the representation of face method in order to search the large database containing visual facial image data more efficiently. And Priority order matching method was used to obtain data that is more reliable. Priority order matching is a system that recognizes faces by giving weight to its most similar facial image. This way we can obtain data that is more reliable and also prevents rejecting reaction from the system. I am proposing a progressive - face recognition algorithm using priority order matching and representation of face, the two method proposed above.

1 Introduction

A facial Recognition System is a computer-driven application that automatically identifies face from still image to motion pictures. It does that by comparing selected facial features in the live image and a facial database. Face recognition extensive use in forensic sciences, biometrics technology is rapidly being adopted in a wide variety of security applications such as computer and physical access control, electronic commerce, digital rights management, background checking, homeland security, and defense. The rapidly expanding research in face recognition processing is based on the premise that information about a user's identity, state, and intent can be extracted from images. There have been a number of approaches in face recognition. Geometric feature based matching,Principal Component Analysis(PCA), Linear Discriminant Analysis(LDA), Neural Network(NN), Support Vector Machine(SVM), and Hidden Markov Model(HMM). Both the PCA and the LDA method are based on statistical approaches and they compare features extracted from face image.In NN based face recognition,

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 790–797, 2007.
© Springer-Verlag Berlin Heidelberg 2007

HyperBF Network and Dynamic Link Architecture(DLA) have been proposed and the DLA method has been recently extend to Elastic Bunch Graph Matching(EBGM). SVM reduces the structural risk to minimize the generalization error through maximizing the margin among classes. HMM classifies a face image utilizing the fact that face features are appeared in sequence such as nose,eye and etc. [1] [2] [3] [4]

In this paper we present results on a face recognition system for real time recognition and reliable recognition rate. Representation of Face Value method was used increase the recognition speed , and Priority Order Matching was used to improve success rates in PCA/LDA Recognition. A Large amount of images are stored in the database in order to raise the recognition success rates. But as the database gets larger, the searching time also increasing accordingly.We propose a method the searches through representative value to solve this problem. And to prevent problems that might occurs because the face is not in a minimum distance, when applying the priority matching method. Because recognition is done by find the nearest distance, priority matching is used. Recognition rate is improved by basic recognition methods using PCA/LDA followed by the priority matching method and representative value.

Section 2 presents a summary of face detection algorithm and the proposed improvement. Section 3 presents we proposed face recognition algorithm which modifies the searching database and increase the recognition rate. Finally Section 4 shows some results which prove the effectiveness of the modifications here proposed on the overall scheme.

Table 1. Typical applications of face recognition

Areas	Representative Works
Biometrics	Drivers Licenses, Entitlement Programs Immigration, National ID, Passports, Voter Registration Welfare Fraud
Information Security	Desktop Log-on(Windows,Linux) Application Security, Database Security, File Encryption Intra-net Security, Internet Access,Medical Records Secure Trading Terminals
Law Enforcement and Surveillance	Advanced Video Surveillance, CCTV Conrol Portal Control, Post-Event analysis Shoplifting and Suspect Tracking and Investigation
Smart Cards	Stored Value Security, User Authentication
Access Control	Facility Access, Vehicular Access
Multimedia management	Face-based search, event detection face-based video segmentation summararization

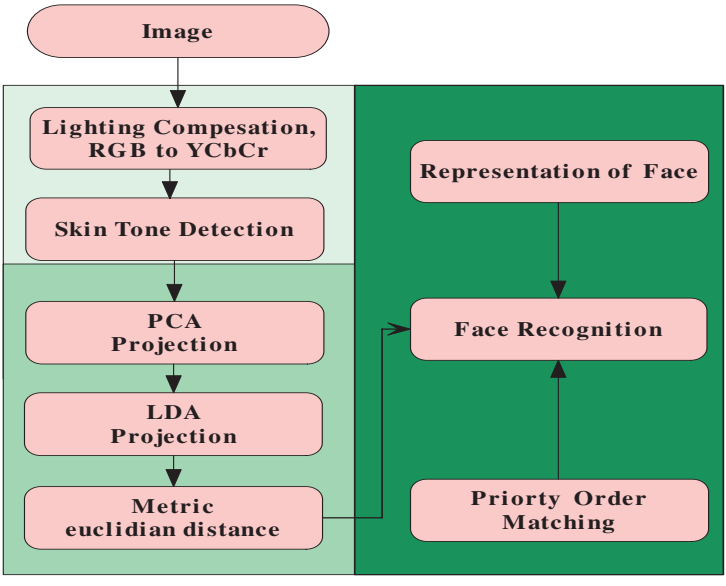


Fig. 1. Block diagram of the proposed face Recognition

2 Preprocessing

2.1 Lighting Compensation

Face detecting depends on the lighting conditions and pose. Lighting compensation is pixels with to top 5 percents of the luminance values in the image as the reference white only if the number of these pixels in sufficiently large. Relative ratio of each of the maximum value is calculated after the mean value of each R,G,B elements inside the white range is calculated. Since the max value of R,G,B is 255 assuming that the mean value of each R,G,B element in the reference white range is Rm, Gm, Bm Maximum value and each of the element ratio is as follows.

$$R_{ratio} = \frac{255}{Rm}, G_{ratio} = \frac{255}{Gm}, B_{ratio} = \frac{255}{Bm} \tag{1}$$

By obtaining a new value by multiplying each of the ratio value with the original R,G,B pixel the visual data is equalized. Visual data is effected by illuminance, but the effect can be minimized by lighting compensation. [5]

2.2 Color Transformation

Generally, various color models are defined by combinations of red, green, and blue primary color components. And the main variation in skin appearance is

largely due to luminance change(brightness), the effect of luminance can be filtered out. Besides RGB color models, there are several other alternative models currently being used in the face detection research. The YIQ and YCbCr color model using in this paper have been applied to face detection frequently.

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ Cb &= -0.16874R - 0.33126G + 0.5B \\ Cr &= 0.5R - 0.41869G - 0.08131B \end{aligned} \tag{2}$$

3 Face Recognition Algorithm

3.1 PCA

PCA is a statistical approach in face recognition. PCA is an optimal transform to compress a signal in the statistical sense. in other words, PCA is used to find a low dimension representation of data. PCA is able to convert R dimension space to N dimension space and reserve main information of original data, so it is easy for further process.the PCA method uses the eigenvector decomposition of the covariance matrix. The data then projected onto the eigenvector corresponding to several eigenvalues to reduce the data to a smaller dimension. [6]

Suppose have N images, see every image as one dimensional vector ($\overline{X_k}, k = 1, 2, ...N_0$) Compute mean vector

$$\overline{X_k} = \frac{1}{N} \sum_{k=1}^{N} \overline{X_k} \tag{3}$$

compute covariance matrix

$$M = \frac{1}{N} [\sum_{k=1}^{N} (\overline{X_k} - \overline{X})(\overline{X_k} - \overline{X})^T] \tag{4}$$

It is well known form matrix theory that the matrix M is positively definite and has only real non-negatively eigenvalues. The eigenvectors of the matrix M form an orthogonal basis for R^d. This basis is called the K - L basis. Let$\{Y_i, i = 1..., d\}$denote the eigenvectors of the matrix M and r_i be the ith eigenvalue. Arrange r_i from larger to smaller and select eigenvectors corresponding to the d' eigenvalues to make projecting space. Project $\overline{X_k}$ into it and make a new vector $\overline{X_k}^{new}$ as the eigenvalue of $\overline{X_k}$. Eigenface is an eigenvector comes from above approach. This approach is provided by Sirovich and Kirby first. Turk and Pentland applied principal component analysis to face recognition and detection. Similar to principal component analysis on training set of face images is performed to generate the Eigenface which span a subspace of the image space[7][8].

3.2 LDA

Linear discriminant analysis easily handles the case where the within-class frequencies are unequal and their performances has been examined on test data generated randomly.This method maximizes the ratio of between-class variance to the within-class variance in any particular data set thereby guaranteeing maximal separability. The within class scatter matrix represents how face images are distributed closely within classes and the between class scatter matrix describes how classes are separated each other. When face images are projected into the optimal discrimination vectors, face images should be distributed closely within classes and should be separated between classes. Let the between-class scatter matrix be defined a

$$S_b = \sum_{j=1}^{c} N_j(\mu_j - \mu)(\mu_j - \mu)^T \tag{5}$$

and the within-class scatter matrix be defined as

$$S_w = \sum_{j=1}^{c} \sum_{x_k \in X_i} (x_i^j - \mu_j)(x_i^j - \mu_j)^T \tag{6}$$

We implement the algorithm for LDA in hopes of providing better classification. The prime difference between LDA and PCA is that PCA does more of feature representation and LDA does data classification. In PCA, the shape and location of the original data sets change when transformed to a different space whereas LDA doesn't change the location but only tries to provide more class separability and draw a decision region between the given classes. This method also helps to better understand the distribution of the feature.[9]

3.3 Representation of Face Component Value

A persons picture, taken in various poses and different light exposure is stored in database to obtain more reliable within changes of pose or light. But most of the time would be wasted searching an individual persons data if data was to be searched all of this. Solving for this problem we propose searching system using mean face. The representative value is first searched after the representative value is stored from an average data of one persons various image differing in exposer and poses. fig-2 is a visual data of the one person image differing in poses and light exposure averaged. Memory that stores the representative value is separately allocated, but can be used for a realtime facial recognition system.

3.4 Priority Order Matching

The persons face is recognized by finding the nearest distance after applying PCA and LDA. But a rejecting reaction might occur if the distance is set to

(a) (b) (c) (d)

Fig. 2. Representation of Face(Mean Face)

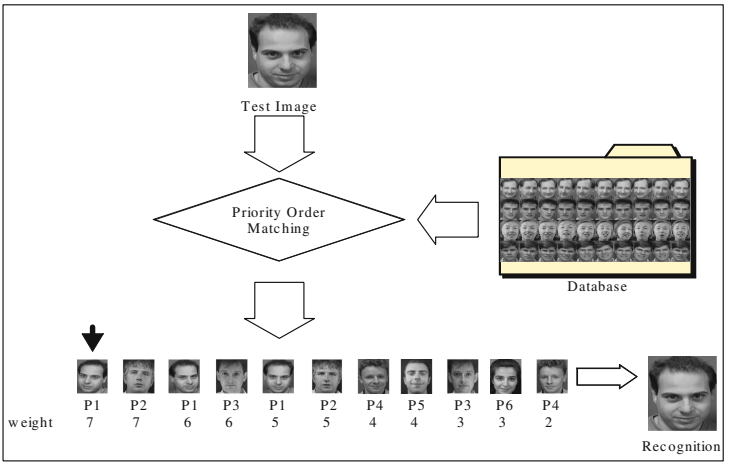


Fig. 3. Block diagram of the proposed face Recognition

critical point. Priority Order Matching weighs top 14 images after sorting it out in closer distance value first image. The face is recognized by finding the largest sum from the duplicated image among the sorted data. Fig-3 represents an outline of our proposed Priority Order Matching. More reliable data can be obtained by applying critical point in this method. And since nearest value can be obtained when no match exists with critical point, the person can be recognized even when rejecting reaction occurs.

4 Experiment Result

The database for performance evaluation is a union of four different database. The Ten face images of each individual werel selected from pose and illumination variation.ORL and UMIST database considered as the pose data set. IMDB database considered as the pose and illumination data set. The whole database

Table 2. Face Database organization for experiments

Database	Total image	Test image	Variation
ORL	400	200	pose variation
UMIST	1000	320	pose variation
IMDB	951	255	pose and Illumination variation

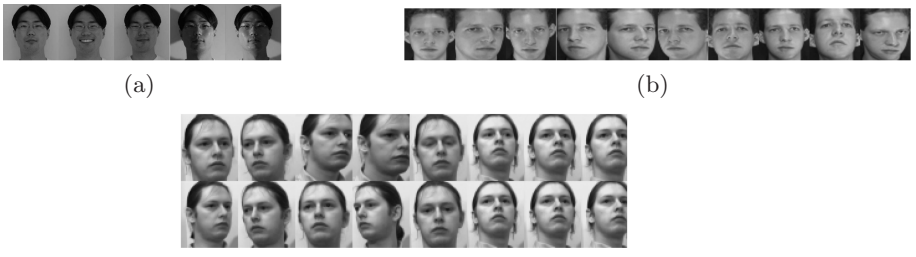


Fig. 4. (a) IMDB (b) ORLDB (c) UMISTDB

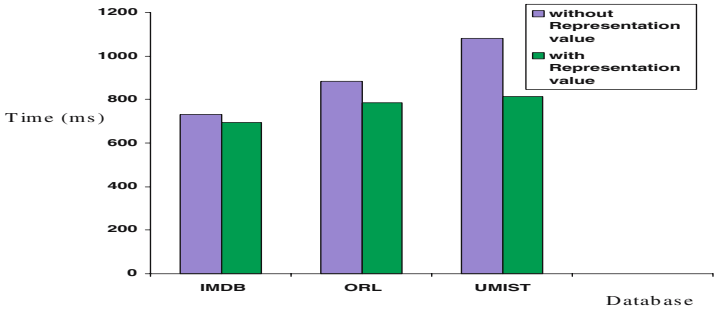


Fig. 5. Result of Recognition Experiments

consists of 2351 images. The images were normalized resized to 46 × 54 pixels.5 pictures of the same person in case of IMDB,10 pictures of the same person in case of ORL DB, 16 pictures of the same person in case of UMIST DB was used for the test.[10][11][12]

The graph shows that better output results compared to existing method as the number of picture of the same person increases. Then, When using Priority Order Matching about 1.5% progress recognition rate was showed compared to the using minimum distance method.

Table 3. Face recognition rate using Priority Order Matching

Databas	Recognition rate	Rejection
Minimum distance method	0.917	0.03
Priorty Order Matching method	0.929	·

5 Conclusion

In this paper, Representation of face value method and Priority Order Matching method for advanced face recognition system is proposed. We represent a face recognition system for real time system by representation of face value method. Representation of Face is efficiency increases as the data of a persons database get's larger Then, get a more reliable recognition rate and reduce recognition rejection use priority order matching method.

We look forward to research on selecting component value of on Representation of face method and the parallel execution of priority order matching.

References

1. Ho, P., Poggio, T.: Face recognition with support vector machine: Global versus component - based approach. In: IEEE International Conference on Computer Vision. vol. 2, pp. 688–694 (2000)
2. Zhao, W., Chellappa, R., Rosenfeld, A., Phillips, P.J.: Face recognition: A literature survey. Technical report, CVL Rechnical Report (2002)
3. Nefian, A.: Hidden markov model based approach for face detection and recognition. PhD thesis, Georgia Institute of Technology (August 1999)
4. Duda, R., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley and sons, Chichester (2001)
5. Hsu, R.-L., Abdel-Mottaleb, jain, A.K.: Face detection in color images. IEEE Pattern Analysis and Machine Intelligence 24, 694–706 (2002)
6. Yambor, W.S.: Analysis of pca-based and fisher discriminant-based image recogniton algorithms. Technical report cs-00-103, Computer Science Department Colorado State University (July 2000)
7. Yang, J., Zhang, D., Frangi, A.F., Yang, J.y.: Two-dimensional pca: A new approach to appearance-based face representation and recognition. IEEE Transactions on pattern analysis And Machine Intelligence 26 (2004)
8. Shen, Y., Chen, Y., Feng, H., He, T.: A recognition algorithm of facial image based on wabelet subbands and decision fusion. Technical report, Zhejiang University of Technology, Hangzhou China (November 2003)
9. Lu, X., Wang, Y., Jain, A.K.: Combining classifier for face recognition. In: International Conference on Multimedia and Expo, vol. 3, pp. 16–19 (2003)
10. KOREA Intelligent Multimedia Laboratory, POSTECH. Imdb (2001)
11. The Digital Technology Group of Cambridge University Computer Laboratory. Orl database of faces (1994)
12. Graham, D., Allinson, N.: The umist face database

Modeling a Legged Robot for Visual Servoing

Zelmar Echegoyen, Alicia d'Anjou, and Manuel Graña*

Computational Intelligence Group, Dept. CCIA
Paseo Manuel de Lardiazábal, 1 20018 San Sebastian - Spain

Abstract. This article presents a contribution to the visual tracking of objects using all the degrees of freedom of an Aibo ERS-7 robot. We approach this issue in a principled way applying ideas of visual servoing. State of the art visual tracking solutions for this kind of robots inspired in the visual servoing approach either are restricted to the head effectors or they apply an inductive learning from experimental data approach to build up the kinematics matrix. In this work we take into account all the effectors which can affect the extrinsic parameters of the robot camera, and therefore in the captured image. We construct the robot kinematic matrix from its description. Visual servoing is performed computing the seudoinverse of this matrix.

1 Introduction

Visual servoing [1] is a technique for robot control which uses as a feedback signal the information extracted from the image sequences taken by one or several video cameras. In fact, it is defined as the control of the end-effector pose relative to a target object or set of features, for robotic manipulators, or the pose of the robot relative to some landmarks, in the case of mobile robotics. A major classification of visual servoing systems distinguishes position-based control from image-based control. In position- based control, the features extracted from the image are used to fit a geometric model of the target and the known camera model to estimate the pose of the robot relative to the target. Control feedback is computed trying to reduce errors in estimated pose space. In image-based servoing, control parameter values are computed on the basis of image features directly. We have chosen the image-based approach to reduce computational delay, avoiding the need for image interpretation and the errors due to sensor modeling and camera calibration. However the image-based approach imposes a linear approximation to obtain the control parameters. This is a significant simplification of the nonlinear and highly coupled robotic system whose effects must be evaluated by physical experimentation. Figure 1 illustrates the main feedback loop in image-based visual servoing with the Aibo.

In the RoboCup robot soccer matches some visual servoing approaches [2,3] have been implemented in the Aibo robot to track the ball. However, these approaches are limited to the movement of the head effectors in order to keep the

* The MEC partially supports this work through grant DPI2006-15346-C03-03.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 798–810, 2007.
© Springer-Verlag Berlin Heidelberg 2007

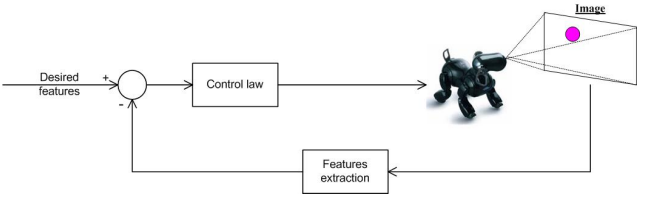


Fig. 1. Visual servoing feedbak loop

ball into the video image. The space in which the ball can be followed is greatly restricted by the robot body pose. Our approach is a systematic application of the visual servoing methodology, aimed at overcoming these limitations.

In this paper we address the problem of maintaining the playing ball in the center of the robot camera image. The only visual feature considered is the center of the ball region as identified in the image by the color detection routines implemented in the robot. We have profited from the CMU's SDK [4] and the SONY's SDK [5]. The image error is the distance in image space between the image center and the centroid of the blob corresponding to the ball. The image features considered are very naive when compared with recent works in other domains (i.e.: [6,7]), however they are the current state of the art in the Aibo environment.

The control parameters are deduced applying the inverse of the linear approximation to the robot kinematic function given by the image Jacobian relative to the considered degrees of freedom. In this paper we detail the construction of the image Jacobian, starting from the geometric specifications of the Aibo robot. The blind application of the control parameter values given by the linear inverse kinematics may move the robot pose out of the useful configuration space, which we define as the set of standing stable positions. These positions are characterized by the relation between the support points and the robot's mass center. The support points are the points of contact of the robots limbs with the support surface. These points may correspond either to the leg ends or to the knees as illustrated in figure 2.

We introduce in section 2 the direct linear kinematics of the robot, then we develop the expression for the inverse kinematics in section 3 and we end up in 4 with some experimental results. The discussion of the physical implementation, the observed robot behavior and future work lines are given in 5

2 Direct Aibo Kinematics

We build the Aibo kinematics as a transformation from the ground supporting plane to the head coordinate system, composing the diverse transformations that correspond to the limbs and head degree of freedom. We start from the supporting points and go up to the head.

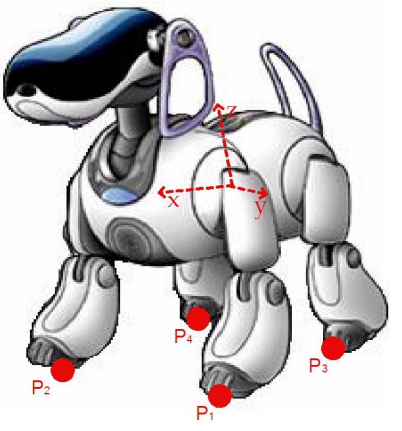


Fig. 2. Points of contact with the supporting surface

As illustrated in figure 2 the robot's feet and the knees are the possible robot support points therefore we need to be able to determine their 3D coordinates at any time.

2.1 Legs Degrees of Freedom

Each leg has three articulations, as shown in figure 4. The legs degrees of freedom are used indirectly towards the support points, so we introduce this concept.

Support Points. The support points are the points of the robot limbs that determine the plane where it is standing on. These points must be determined in the coordinate system of the dog. From the point of view of the center of the robot body the supporting plane apparently varies when the robot servos are affected when the physical reality is that the plane remains fixed and the robot changes its pose. We use the robot body center of mass because the Aibo possesses an inertial sensor than gives us feedback on the motion of this point.

Each leg has a unique support point that can be the foot as well as the knee, and, according to the restriction that the robot must be standing, at least three of the legs must have their supporting points in contact with the ground; therefore there are 32 possible support planes if we take into account all feasible combinations support points that may give us a standing configuration of the robot. In order to determine which combination of supporting points coincides with the physical supporting surface we obtain the plane equation for every possible combination.

For a given combination of support points we have the plane equation $\pi : ax + by + cz + d = 0$, then we evaluate to which hemisphace belong the points that have not been taken into account to build the plane equation; if for any one of them we find $(x_p, y_p, z_p)^T : ax_p + by_p + cz_p + d < 0$, it means that this support point is under the plane and therefore this plane is not the ground

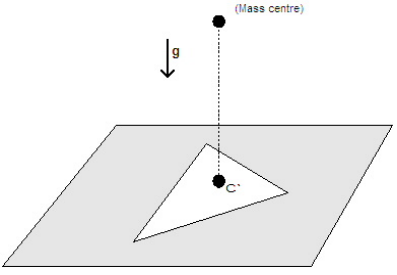


Fig. 3. Condition for supporting points on the ground plane

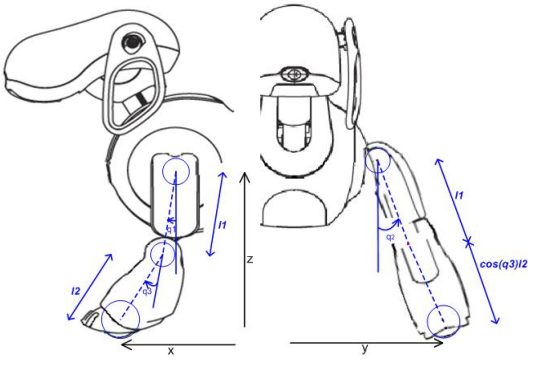


Fig. 4. Geometry of the leg articulations

surface. Besides, in order for the robot to be standing in a stable pose, the projection of the body center of mass must lie inside of the triangle defined by the three supporting points in contact with the ground surface. This condition is illustrated in figure 3. Therefore, the search for the ground support points is guided by testing this condition on each triplet of leg supporting points. For those triplets that meet the condition, we fit the plane equation and test that the remaining supporting points remain above this plane.

Feet and Knees Positions. In order to obtain the ground supporting plane, is necessary to determine which are the supporting point coordinates for each leg in the reference space centered on the robot body center of mass. It is necessary to determine the positions of the feet and knees in function of the articulation states, given by their torsion angles.

We find the foot center position, for the front left leg, using the following coordinate system transformations.

T_1: Translation along de z-axis of length $-l_1$.
R_1: Clockwise rotation about y-axis by angle q_1.
R_2: Counterclockwise rotation about x-axis by angle q_2.

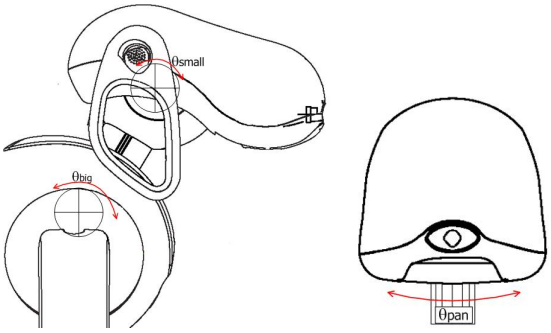


Fig. 5. Geometry of the head articulations

R_1: Clockwise rotation about y-axis by angle q_3.
T_2: Translation along de z-axis with length $-l_2$.
T_l: Translation along de x-axis with length $\frac{1}{2}l$, being l the robot length.
T_a: Translation along de y-axis with length $\frac{1}{2}a$, being a the robot width.

In homogeneous coordinates the transformation from the body center to the foot coordinate system can be described as the product of transformation matrices:

$$\begin{pmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{pmatrix} = (R_1.R_2.T_1.R_3.T_2). \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

This equation is valid for the robot front left leg; however, due to the symmetry of leg coordinate systems, only a few signs must be changed to get the positions of the other three leg's feet. In order to find the coordinates of each knee in the body reference system we only have to do the three first and the two last transformations used to determine the foot coordinates:

2.2 Head Degrees of Freedom

The Aibo ERS-7 has three degrees of freedom in the head. That introduces ambiguity in the control trajectories needed to track the ball trajectory. Figure 5 shows the two tilt degrees of freedom of the Aibo, denoted θ_{big} and θ_{small}. The first head tilt degree of freedom corresponds to the neck base pivoting along part of the dog chest, while the second one allows the head to move vertically using as the rotation center the joint between the neck and the head. The third degree of freedom, called θ_{pan}, allows a perpendicular rotation to the previous one, moving the head from side to side.

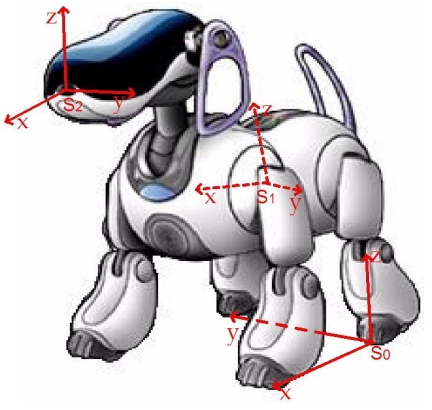


Fig. 6. Reference systems involved in the visual servoing

2.3 Image Features

The stated goal is to bring the ball in the image center, so the target features are the image center coordinates and the observed features from the real world are the coordinates of the ball region center and its diameter. But these features must be expressed in terms of the robot degrees of freedom, in order to use the Jacobian to determine the feature sensitivity respect to each articulation positions changes.

Coordinate Reference Systems. In order to obtain the ball position expressed in the S_0 system base it is necessary to obtain the transformation matrices between the different systems.

These reference systems are illustrated in figure 6.

Transformation between S_0 and S_1. In order to define the coordinates changes between the base system, S_0, and the body system, S_1, we define the S_0 vectors in the system S_1, and then do the translation between them. So, we separate the transformation in rotation and translation, although it exists an scale component.

The entire transformation uses the supporting points positions: $r_i = (x_i, y_i, z_i, 1)^T$, $r_j = (x_j, y_j, z_j, 1)^T$, $r_k = (x_k, y_k, z_k, 1)^T$.We use the position point r_i as the origin of S_0, and the vectors $\overrightarrow{r_i r_j}$ and $\overrightarrow{r_i r_k}$ as the two first vectors, and we built the third vector as the vectorial product of the two first.

So we built the rotational matrix, R, from the three vectors of S_0

$$R = \begin{pmatrix} \downarrow & \downarrow & \downarrow & 0 \\ r_j - r_i & r_k - r_i & \langle r_k - r_i, r_j - r_i \rangle & 0 \\ \downarrow & \downarrow & \downarrow & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

and we define the translational matrix from the origin of S_1 to the origin of S_0,

$$T = \begin{pmatrix} 1\ 0\ 0\ x_i \\ 0\ 1\ 0\ y_i \\ 0\ 0\ 1\ z_i \\ 0\ 0\ 0\ 1 \end{pmatrix}. \tag{2}$$

So, composing the two transformations we finally obtain the matrix change from S_0 to S_1,

$$_{S_1}I_{S_0} = T \cdot R. \tag{3}$$

Transformation between S_1 and S_2. The transformation between these systems can be done through the compositions of more elemental transformations. We will compose the transformations that go from the body system S_1, to the head system S_2.

The first transformation is a translation from the camera base to the top of the neck, T_1. Next, we have to rotate the head, taking into account the nod and pan articulations, we call this rotational matrix R_1. Then, we have to use the tilt articulation defining the rotational matrix R_2. Finally, the translation T_2, between the neck base and the body center, take the system to the S_1 origin.

The result of the matrix composition is the transformation between the systems S_1 and S_2

$$_{S_2}I_{S_2} = T_2 \cdot R_2 \cdot R_1 \cdot T_1. \tag{4}$$

Observed Image Features. The camera reference system is fixed to the robot head, and define the ball position according to the vision camera of the robot.

The observed image features, $c = (u, v)^T$, are determined by the ball position in the camera system, according to the following relation

$$\begin{pmatrix} u \\ v \end{pmatrix} = f(b_2) = \frac{\lambda}{x_{b_2}} \begin{pmatrix} y_{b_2} \\ z_{b_2} \end{pmatrix}. \tag{5}$$

The features are expressed in terms of the ball position in the system S_2, but as we had supposed the ball was fixed respect to S_0, we could obtain the features expressed in function of the head robot articulations and the support points positions, using the ball position in S_0 and the transformation between S_0 and S_2.

$$\begin{pmatrix} u \\ v \end{pmatrix} = f(_{S_2}I_{S_0}.b_0) \tag{6}$$

2.4 Feature Jacobian Matrix

Now we will construct the Jacobian matrix that relates the variations of the diverse degrees of freedom of the robot with the variations in the image plane.

Dependence on the Features. Deriving the equation 5 we get the following relation:

$$\begin{pmatrix}\delta u\\ \delta v\end{pmatrix} = \begin{pmatrix}\frac{-\lambda.y_p}{x_p^2} & \frac{\lambda}{x_p} & 0 & 0\\ \frac{-\lambda.z_p}{x_p^2} & 0 & \frac{\lambda}{x_p} & 0\end{pmatrix}\begin{pmatrix}\delta x_{b_2}\\ \delta y_{b_2}\\ \delta z_{b_2}\\ 0\end{pmatrix} \tag{7}$$

We call J_{cb} the Jacobian matrix of the equation 7. Then J_{cb} defines a linear transformation from variations of the positions of the ball in S_2 into variations of the image features.

$$\Delta c \simeq J_{cb} \cdot \Delta b \tag{8}$$

Dependence on the Target Object. We saw that the ball position in the camera system could be expressed as a function of the support points and the head robot articulations, equation 6. By deriving this equation we get the Jacobian matrix that relates the variations in the image ball position with the variations in the support points positions and in the head articulations

$$J_{br} = \frac{\delta({}_{S_2}I_{S_1} \circ {}_{S_1}I_{S_0})}{\delta r} b_0 \tag{9}$$

Using the chain rule, we rewrite the equation 9:

$$J_{br} = [\frac{\delta({}_{S_2}I_{S_1})}{\delta r} \circ ({}_{S_1}I_{S_0}) + ({}_{S_2}I_{S_1}) \circ \frac{\delta({}_{S_1}I_{S_0})}{\delta r}]b_0 \tag{10}$$

As ${}_{S_2}I_{S_1}$ is a function of r_{head} (head articulations) and ${}_{S_1}I_{S_0}$ is a function with parameter r_{legs} (support points positions), we rewrite 10:

$$J_{br} = [\frac{\delta({}_{S_2}I_{S_1})}{\delta r_{head}} \circ ({}_{S_1}I_{S_0}) + ({}_{S_2}I_{S_1}) \circ \frac{\delta({}_{S_1}I_{S_0})}{\delta r_{legs}}]b_0 \tag{11}$$

The dependence between the variations in the ball position and the variations in the head degree of freedoms and in the legs positions can be summarized by:

$$\Delta b \simeq J_{br} \cdot \Delta r \tag{12}$$

Dependence on the Support Points. The next step is obtaining a linear transformation between the variations of the legs degrees of freedom and the ground support points coordinates in the body reference system.

First we observe that according to which part of the leg is in contact with the ground there are two possible jacobean matrices, one for the foot (J_{p_i}) an another for the knee (J_{r_i}). We model the changes in the foot and the knees coordinates according to the degrees of freedom variations, using the Jacobians as follows:

$$\Delta p_i \simeq Jp_i \cdot \Delta J_i \tag{13}$$

$$\Delta r_i \simeq Jr_i \cdot \Delta J_i \tag{14}$$

Being ∂J_{3i}, ∂J_{1i} and ∂J_{2i} the value of the variations of the degrees of freedom in leg i.

Composing with the support points Jacobian at every moment, we obtain the following jacobian matrix:

$$\begin{pmatrix} \partial P_1 \\ \partial P_2 \\ \partial P_3 \\ \partial P_4 \end{pmatrix} = \begin{pmatrix} M_1 & 0 & 0 & 0 \\ 0 & M_2 & 0 & 0 \\ 0 & 0 & M_3 & 0 \\ 0 & 0 & 0 & M_4 \end{pmatrix} \begin{pmatrix} \partial J_1 \\ \partial J_2 \\ \partial J_3 \\ \partial J_4 \end{pmatrix} \tag{15}$$

The jacobian matrix receives the name $J_{p\theta}$,where M_i is:

- Jp_i, if the support point for the leg i is the foot.
- Jr_i, if the support point for the leg i is the knee.
- $Zero$ (the matrix with all the elements equal 0) if this leg has not a lean point on the plane.

The dependence of the support ground points on the limb's degrees of freedom is summarized as follows:

$$\Delta P \ \simeq \ J_{p\theta}.\Delta J \tag{16}$$

Dependence on the Robot Articulations. The following matrix relates the variations in the generic support points, δr, with the variations in the true legs support points and the free leg support point, δp. We call δr_h and δp_h the head articulation variations in order to define r and p.

$$\begin{pmatrix} \delta p_h \\ \delta p_1 \\ \delta p_2 \\ \delta p_3 \\ \delta p_4 \end{pmatrix} = \begin{pmatrix} Id & 0 & 0 & 0 \\ 0 & M_{1i} & M_{1j} & M_{1k} \\ 0 & M_{2i} & M_{2j} & M_{2k} \\ 0 & M_{3i} & M_{3j} & M_{3k} \\ 0 & M_{4i} & M_{4j} & M_{4k} \end{pmatrix} \begin{pmatrix} \delta r_h \\ \delta r_i \\ \delta r_j \\ \delta r_k \end{pmatrix} \tag{17}$$

We call $J_{pr} \in \mathbb{R}^{15\times 12}$ the Jacobian matrix of equation 17, and we define this matrix as a composition of matrixes of size 3×3. The first row starts with the identity and the rest of matrixes are null. So, δr_h and δp_h are equals.

To the rest of matrixes:

$M_{xy} = Id$, if the leg x has the support point y,
$M_{xy} = 0$, if the leg x has a support point different from y,
$M_{xy} = \alpha_{xy}$, if the leg x has not a generic support point.

So, if the leg x has a generic support point, then the row x has a identity matrix and three null matrix; while for the free leg we get the following row

$$\begin{pmatrix} 0 & \alpha_i & \alpha_j & \alpha_k \end{pmatrix} \tag{18}$$

This coefficient matrixes define the relation between the free leg variations and the variations in the rest of legs. This coefficients can be used to generate stability

and advance behaviors. If the first element of the row is not 0, then we can make the free leg position depend on the head articulations variations.

The dependence between the variations in the legs articulations with the supporting points positions and the head articulations variations can be summarized in the following equation

$$\Delta p \simeq J_{pr} \cdot \Delta r \tag{19}$$

Defining the Jacobian matrix $J_{r\theta}$, as the product $J_{pr}^{+} \circ J_{p\theta}$, we get the following dependence relation between the variations of support points positions and robot articulations,

$$\Delta r \simeq J_{r\theta} \cdot \Delta\theta \tag{20}$$

Full Jacobian Matrix. Finally, to obtain the full Jacobian matrix that models the dependence of the image features on the diverse degrees of freedom of the robot we must compose the previous transformations given by equations (8), (12), (19) and (16).

$$\Delta c = [(J_{cb} \circ J_{br}) \circ J_{r\theta}].\Delta\theta \tag{21}$$

We denote the composite matrix $J_{c\theta} = (J_{cb} \circ J_{br}) \circ J_{r\theta}$.

3 Inverse Kinematics

The goal of the stated visual servoing problem is to determine the instantaneous variation at each robot degree of freedom that will be needed to bring the ball center to the image plane center.

In order to determine the velocity at each robot degree of freedom we should obtain the inverse of the $J_{c\theta}$ matrix in equation 21. However, this is not possible because the matrix is not invertible. As we have more degrees of freedom than image features, the problem is overconstrained, because there are not sufficient features to determine the movements in an unique way.

The general solution is to use the pseudoinverse of $J_{c\theta}^{+}$, thus obtaining the following approximation to the instantaneous robot degree of freedom velocity vector:

$$\dot{\theta} = J_{c\theta}^{+}\dot{c} + (I - J_{c\theta}^{+}J_{c\theta})n \tag{22}$$

Being n an arbitrary vector of R^{15}.

In general, $(I - J_{xj}^{+}J_{xj})n \neq 0$, and all the vectors of the form $(I - J_{xj}^{+}J_{xj})n$ belong to the kernel of the transformation associated to $J_{c\theta}$. This solution minimizes the norm

$$\left\| \dot{x} - (J_{c\theta})\dot{\theta} \right\| \tag{23}$$

As our objective is to bring the ball to the image center, we can neglect the second order term, so our estimation of the velocity vector will be

$$\dot{\theta} = J_{c\theta}^{+}\dot{c} \tag{24}$$

This solution does not take into account the restriction of keeping the distances between supporting points constant. So, we need to determine how the variations in the supporting points positions, as a consequence of the robot joint movement, affect the distances between them. Let us consider the vector of distances:

$$l = \begin{pmatrix} \|l_1\| \\ \|l_2\| \\ \|l_3\| \end{pmatrix} = \begin{pmatrix} \|r_i - r_j\| \\ \|r_j - r_k\| \\ \|r_k - r_i\| \end{pmatrix} \tag{25}$$

Differencing l we get the Jacobian matrix J_{lr} that relates these changes in position with the change in relative distances between supporting points:

$$J_{lr} = \begin{pmatrix} 0 & \frac{\delta l_1}{\delta r_i} & \frac{\delta l_1}{\delta r_j} & \frac{\delta l_1}{\delta r_k} \\ 0 & \frac{\delta l_2}{\delta r_i} & \frac{\delta l_2}{\delta r_j} & \frac{\delta l_2}{\delta r_k} \\ 0 & \frac{\delta l_3}{\delta r_i} & \frac{\delta l_3}{\delta r_j} & \frac{\delta l_3}{\delta r_k} \end{pmatrix} \tag{26}$$

Finally this dependence is summarized in the following equation:

$$\Delta l \simeq J_{lr} \cdot \Delta r \tag{27}$$

Now we may combine the following Jacobian matrices: J_{rc}, J_{lr} and $J_{r\theta}$ to compute the variations on the robot articulations that make the image features converge to the desired features, keeping constant the distances between the supporting points. To fulfill this restriction, the Δr vector must belong to the kernel of the transformation associated to J_{lr}. The following equation ensures that Δr belongs to the kernel of J_{lr}.

$$\Delta r = [(I - J_{lr}^{+} J_{lr})\{(I - J_{lr}^{+} J_{lr}) J_{rc}^{+}\}^{+}]\Delta c \tag{28}$$

As our final objective is to compute the movements at the robot joints, we multiply by the seudoinverse of $J_{r\theta}$. The control of the robot joints bringing the ball center to the image center is given by the following iterative rule:

$$\Delta\theta = \alpha J_{r\theta}^{+}(I - J_{lr}^{+} J_{lr})\{(I - J_{lr}^{+} J_{lr}) J_{rc}^{+}\}^{+} \Delta c, \tag{29}$$

where α is the control gain that modulates the application of the rule, the experimental results show the sensitivity of the approach to the values of this gain.

This equation is unrestricted and may drive the robot into unstable configurations, that is, to articulation configurations that fall out of the region of stable poses in configuration space. Stable poses are characterized by the existence of a triplet of ground support points which fulfill the condition illustrated in figure 3. When this does not happen, or the projection point is too close to the triangle boundary, we restrict the visual servoing to the head degrees of freedom, using the transformation $S_0(I)S_2$ to construct a reduced Jacobian M_h that relates the image features to the head degrees of freedom. Its seudoinverse gives the control

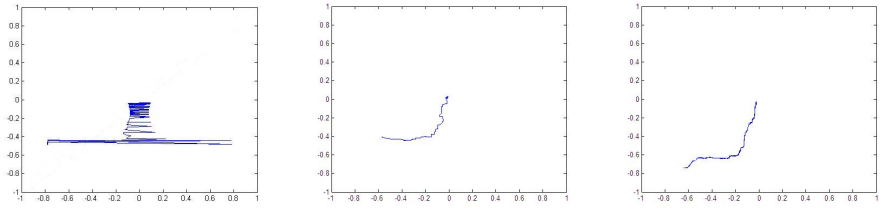
for the head degrees of freedom. This reduced approach has already been applied in [2,3]:

$$d\theta = M_h^+ . dr \tag{30}$$

4 Experimental Results

We tested our approach on the Aibo robot. In the real life experiments the starting position of the ball is in the lower left corner of the image. The experimental results show the sensitivity of our control rule convergence to the gain parameter. Figure 7 show some instances of trajectories of the ball center in the image plane obtained for different settings of the gain parameter. It can be observed that $\alpha = 1$ produces a fast convergence followed by strong oscillations. Lower vales equivalent to damped control, show slower but oscillation free convergence. Very small values result in control commands below the precision of the robot joints.

The experiments have also show that there is some sensitivity to the initial robot pose. In order to prevent the robot to escape its standing configuration region, if we detect that the robot tries to perform a joint movement that risks escaping the standing configuration region, we block all motion of the body joints and restrict the calculation of the motion of the head joints, as described above.



(a) (b) (c)

Fig. 7. Trajectories in the image plane of the ball center for different values of the gain parameter (a) $\alpha = 1$, (b) $\alpha = 0.5$, (c) $\alpha = 0.3$

5 Conclusion

We have developed the visual servoing for the whole set of degrees of freedom of the Aibo 7 following a principled approach. From the geometrical description of the robot we have constructed the full Jacobian matrix that linearizes the functional dependence of the image plane viewed by the robot camera on the robot degrees of freedom. The seudoinverse of this Jacobian matrix provide the desired controls. The blind application of this control strategy may lead the robot to unstable or unfeasible configurations for a standing pose. Therefore, we test the stability of the robot configuration. When it is compromised we restrict the

visual servoing to the head. The implementation shows that the approach gives real time response when the seudoinverse is computed in the onboard processor of the robot. We are actually performing the real time experiments and collecting performance information.

References

1. Hutchinson, S., Hager, G.D., Corke, P.I.: A tutorial on visual servo control. IEEE Transactions on Robotics and Automation 12(5), 651–670
2. Quinlan, M., Murch, C., Moore, T., Middleton, R., Li, L., King, R., Chalup, S.: The 2004 NUbots Team Report (2004), http://robots.newcastle.edu.au/publications/NUbotFinalReport2004.pdf
3. Röfer, Th., Burkhard, H.-D., Düffert, U., Hoffmann, J., Göhring, D., Jüngel, M., Lötzsch, M., Stryk, O.v., Brunn, R., Kallnik, M., Kunz, M., Petters, S., Risler, M., Stelzer, M., Dahm, I., Wachter, M., Engel, K., Osterhues, A., Schumann, C., Ziegler, J.: GermanTeam RoboCup 2003. Technical report (2003), http://www.robocup.de/germanteam/GT2003.pdf
4. www.tekkotsu.org
5. Sony Corporation, OPEN-R SDK Model Information for ERS-7 (2003)
6. Pomares, J., Torres, F.: Movement-flow-based visual servoing and force control fusion for Manipulation Tasks in unstructured environments. IEEE trans. Systems, Man and Cybernetics, Part C 35(1), 4–15
7. Garcia-Aracil, N., Malis, E., Aracil-Santonja, R., Perez-Vidal, C.: Continuous visual servoing despite the changes of visibility in image features. IEEE Trans. Robotics 21(6), 1214–1220

Information Extraction in a Set of Knowledge Using a Fuzzy Logic Based Intelligent Agent

Jorge Ropero, Ariel Gómez, Carlos León, and Alejandro Carrasco

Department of Electronic Technology, University of Seville, Spain
jropero@dte.us.es, ariel@us.es, cleon@us.es, acarrasco@us.es

Abstract. A method for Information Extraction (IE) in a set of knowledge is proposed in this paper in order to answer to user consultations using natural language. The system is based on a fuzzy logic engine, which takes advantage of its flexibility for managing sets of accumulated knowledge. These sets can be built in hierarchic levels by a tree structure. A method of consultation based on a fuzzy logic application provided with an interface that one may interact with in natural language is also proposed. The eventual aim of this system is the implementation of an intelligent agent to manage the information contained in an internet portal.

Keywords: Fuzzy Logic, Intelligent Agent, Information Extraction, Vector Space Model.

1 Motivations

The approach to the contents of an extensive set of accumulated knowledge –a database, a summary of documents, web contents, goods in a store, pictures, etc – is an important concern nowadays. Information Retrieval (IR) deals with large collections of textual material, and its aim is to satisfy user queries and needs [1]. These needs meet increased when the matter is not in the form of text, the user in question is not a habitual of the matter, there are ambiguous contents, bad organization or, simply, complex topics or a great amount of information difficult to manage.

Eventually, unsuccessful attempts can turn out to be frustrating if the exact term or terms are not used to make the consultations - a machine only will answer adequately if it is asked in an exact way -, and one can eventually end in a paradox: the less one knows the more difficult it is to find the answers. In many cases the solution is to seek help from an expert on the topic. In fact the person asked to help is an interpreter who is able to generate a syntactically and semantically a correct search obtaining the desired answers. Consequently, there is the need for an agent to interpret the vague information we provide, giving us concrete answers related to the existing contents of the set of knowledge. This should be based on an estimation of the certainty of the relation between what we have expressed in natural language and the contents stored in the set of knowledge.

To solve this, this paper proposes a method to extract information from a web portal. This method is founded on a Vector Space Model (VSM) based classification

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 811–820, 2007.
© Springer-Verlag Berlin Heidelberg 2007

of contents by creating a few indexes based on key words, and a method of consultation based on a Fuzzy Logic (FL) application with an interface that one may interact with in natural language. We then propose an artificial intelligence (AI) application based on the use of FL.

2 Related Work

As said above, the quantity of web information grows exponentially. For a long time researchers have faced the problem of managing this high amount of information IR research deals predominantly with documents. Achieving both high recall and precision in IR is one of its most important objectives. IR has been widely used for text classification [2, 3] introducing approaches such as Vector Space Model (VSM), K nearest neighbour method (KNN), Bayesian classification model, neural networks and Support Vector Machine (SVM) [4]. VSM is the most frequently used model.

Moreover, the use of unlabeled, distributed and heterogeneous web data has prompted the need for developing automatic mining techniques on the World Wide Web, giving rise to the term web mining [5]. Many techniques like Web Content Mining (WCM), Web Structure Mining (WSM) or Web Usage Mining (WUM) have been used in the last years [6]. WCM and IR are separated by a thin line [5]: they are both document finding oriented. The main difference between them lies on the use of semistructured data on the web. This makes WCM especially challenging [7]. WSM deals with web structure itself. WSM techniques enable improving web contents or extracting knowledge and other required information. The major methods of Information Extraction (IE) involve writing wrappers [5]. Some examples of the use of wrappers for IE are STAVIES, which presents a fully automated IE method for Web pages [8] or OMINI [9].

What we propose is a fully novel method using Fuzzy Logic (FL) not only to extract the required information but making the most of FL flexibility to extract related information too, giving answers to user consultations in natural language and bearing in mind that non-expert users tend not to be exact in their searches.

3 Mode of Operation

3.1 Objectives

The main objective of the system designed must be to let the users find possible answers to what they are looking for in a huge set of knowledge. With this aim, the whole set of knowledge must be classified into different objects, as shown in Figure 1. These objects are the possible user consultations, organized in hierarchic groups. A standard question is assigned for every object, and different key words from each standard question must then be selected in order to differentiate one object from the others. Finally, term weights are assigned to every word for every level of hierarchy in a scheme based on a vector space model. These term weights are the inputs to a fuzzy logic system, which must detect the object to which the correspondent user consultation refers. Term weighting scheme is shown in Figure 2.

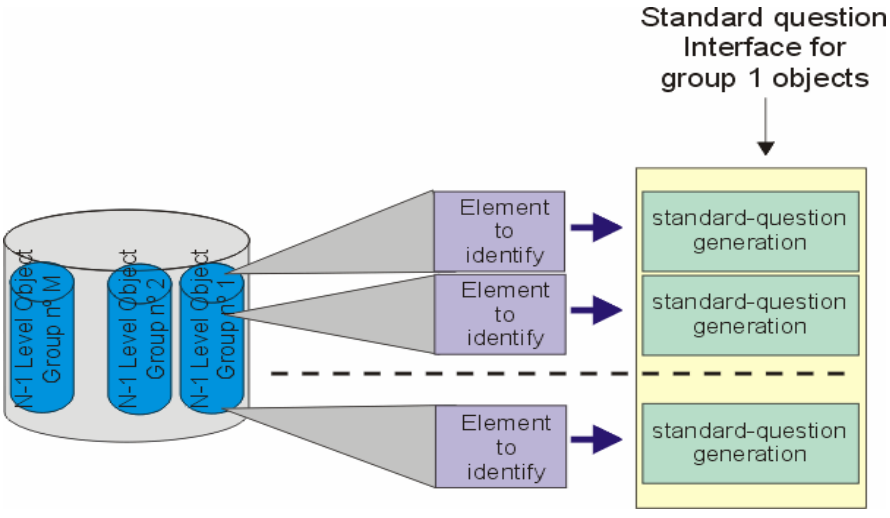


Fig. 1. Generated standard question interface

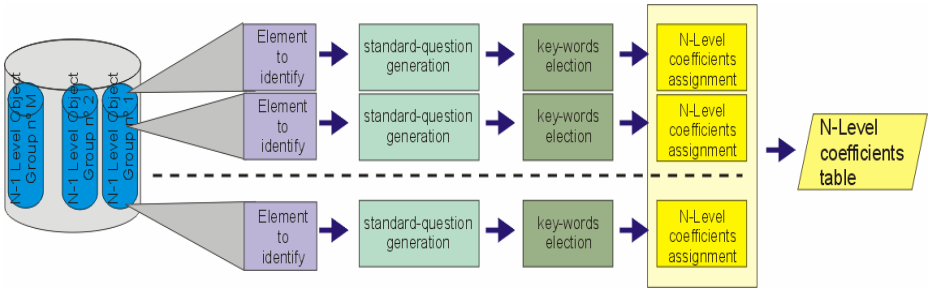


Fig. 2. Term weighting scheme

3.2 Hierarchic Structure

The standard-questions collection generated from the whole set of knowledge must be organized in hierarchic groups. This offers the advantage of easier handling. For tests, we decided to use the questions-answers database of the University of Seville to create a list of the most frequently asked questions. The whole knowledge was organized into three levels: topic, section and question. For any web page, it would be enough to create a bank of possible questions–answers, arrange them hierarchically, and identify them as objects.

Every possible user question must be related with standard questions in order to present its standard answers as possible answers to this user question. This may be seen in Figure 3. There are obviously many ways of asking, so the aim of the system is to identify real user consultations and what we have called standard questions. As there may be several questions similar to the one asked by the user or as it may be interesting for the user to get related answers, several answers will be presented.

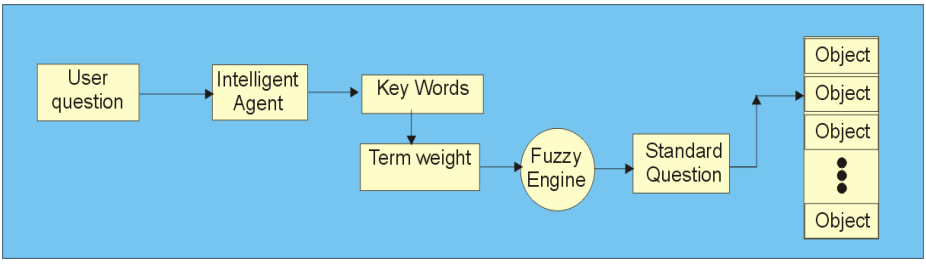


Fig. 3. Mode of operation

When a question is made, the first processing step consists of distinguishing the key words in the consultation. These words must be searched in a database which must contain all the words that are related somehow to the content of the subject we are dealing with –see Figure 4. Another database with the possible answers becomes necessary.

Tema1Nivel2Apdo2 : Tabla

Campo1	Campo2	Campo3	Campo4
colegiados	0,8	0	0
equipo	0	0,5	0
gobierno	0	0,8	0
organos	0,9	0	0
puestos	0	0	0,7
relacion	0	0	0,7
sevilla	0,2	0,2	0,2
trabajo	0	0	0,7
universidad	0,3	0,3	0,3

Registro: 1 de 9

Fig. 4. Index words database

Key words are assigned to every standard question in order to identify it. These words are chosen among those that could appear in a possible consultation. As mentioned above, the whole knowledge is grouped in various hierarchic levels, so the belonging of these words to different levels is determined by a few numerical coefficients indicating how significant the considered word is within the level in question. With this aim, weight vectors are assigned to each word. Each vector contains the certainty of belonging to every fuzzy set. It is important to notice that the same word can belong to several different sets.

3.3 Term Weighting

As the system must work using a natural language interface, we propose the use of a system based on VSM for term weighting. Automatic document indexing is usually based on the frequency of occurrence, that is to say, words with a high frequency of occurrence are less significant than words with a low one, and the inverse document

frequency, that is, how often it occurs in the whole document. The main limitation of this scheme is created by terms which are unhelpful for identification but have the same frequency of occurrence as the relevant ones [3,4,10-14].

Thus, we propose a novel scheme based on a vector space model which also takes into account if the word is important for the meaning of the question itself or if the word is linked to other words and its relationship with the whole set of knowledge for term weighting [3,15].

The success of the proposed method depends to a great extent on a correct assignment of the coefficients to the key words, that is, the building of the weight vectors. The process consists of 3 stages: election of key words, coefficient assignment of the chosen words; and modification of the index values in order to obtain the desired minimal certainties.

Election of key words. As mentioned earlier, for every element that has to be determined, a question is assigned in natural language. These questions are called *standard questions*. From every standard question key words are chosen. These key words will allow the user consultation and these standard questions to be matched and, therefore, to indicate the required element. Word election is based on its *concretion,* meaning the degree of relation of the word with the element to be identified. This excludes articles, conjunctions, verb forms, etc, unless they are strongly significant for the question structure.

Coefficient assignment. For every key word, coefficients corresponding to every level of hierarchy of the whole set of knowledge must be assigned. The higher the relationship between the present level and the key word, the higher the coefficient of that key word will be for that level. These coefficients are the inputs to every fuzzy logic system, as described in Section 2.4.

Standard question recognition test. Once the coefficient assignment is made, standard question recognition tests must take place. These tests are based on using standard questions as user consultations.

Coefficient modification. The results obtained for standard question recognition tests represent the relationship of every standard-question relationship with itself. The higher it is the better the results are. However, it must be taken into account that similar questions must have a high degree of certainty too. This allows for the modification of coefficients in order to achieve better results. An example of how we have modified some coefficients is shown in Section 3.1.

3.4 Fuzzy Logic System

Fuzzy logic arises as a response to the inflexibility of the classic binary logic [16-17]. By means of a set of functions, a degree of flexibility may be given to these epithets: what may be cool for a Sevillian might be mild for a Berliner.

A fuzzy logic system gives flexibility for term weighting. More important than having a concrete value for weights, what really matters is that a feature is represented by a word. It is not so important that a weight is 0.8 or 0.9: the weight is HIGH in both cases.

All the key words are extracted for comparison with the ones contained in our key word database. As mentioned earlier, the whole set of knowledge is arranged in levels of hierarchy. The inputs to the fuzzy logic system are the coefficients of belonging to every level. The set "belonging to every level 1" is analyzed bearing in mind the value returned by the fuzzy engine. If the level of certainty is lower than a predefined value, the content of the corresponding set is rejected. Starting from level one and using a tree structure makes it possible to reject a great amount of content which will not be considered in future searches.

For every set that has overcome a minimum certainty threshold, the process is repeated and the coefficients of belonging corresponding to every level 2 set are evaluated. Sets where the degree of certainty returned by the fuzzy engine does not surpass a certain minimal threshold are rejected. If they surpass the threshold, the method for determining the belonging to the following level is applied to them. This process is repeated until the last level. The answers correspond to those last level elements in which certainty has overcome the definite threshold. There can be more than one answer. The vaguer the questions, the more answers we will obtain. This process may be seen in Figure 5.

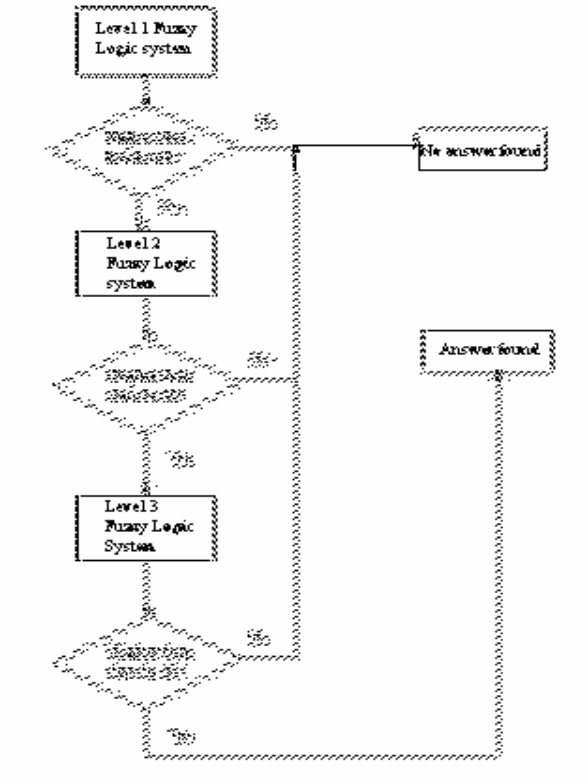


Fig. 5. Fuzzy logic system: operation mode

The heart of the fuzzy logic system is the fuzzy engine. This engine is responsible for determining the probability that the key words contained in a consultation will belong to a certain fuzzy set in a specific level. The engine must evaluate the belonging to every set for the corresponding level. Thus, the engine takes the coefficients of the key words for that set as inputs. The fuzzy engine output will be determined by the defined rules. These rules are of the IF … THEN type. An example of a rule might be this:

IF word_index1 is HIGH AND word_index2 is MEDIUM AND word_index3 is LOW, THEN output is HIGH.

3.5 System Administrator

The system requires a system administrator who has basically three functions:

a) Defining and modifying term weights and rules.
b) Adding new words to the database when necessary.
c) Creating a system feedback which asks the eventual users for their opinion about the answers given by the assistant in order to take the necessary steps in each case.

Thus far, an administrator interface has been created – Figure 6 - .

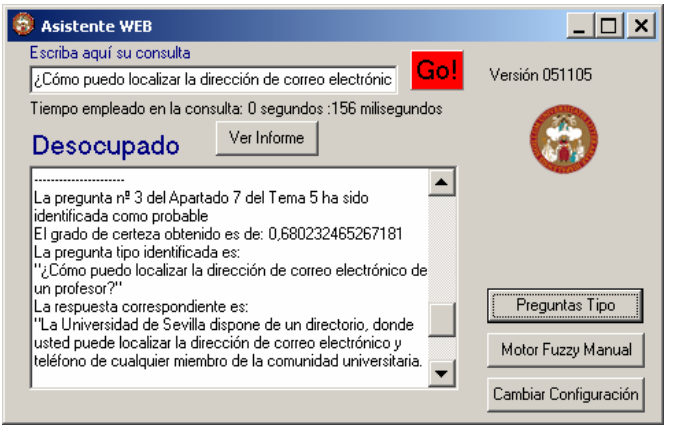


Fig. 6. Administrator interface

4 Tests and Results

4.1 Test Definition

Tests are based on the use of standard questions as user consultations. The first goal of these tests is to check that the system makes a correct identification of standard questions with an index of certainty higher than 0.7. The use of fuzzy logic makes it possible to identify not only the corresponding standard question but others as well. This is related to *recall*, though it does not match that exact definition [18]. The

second goal is to check if the required standard question is among the three answers with higher degree of certainty. These three answers should be presented to the user. The correct answer must be among these three options. This is related to *precision*, though it does not match that exact definition either.

To achieve these goals, it will be necessary to modify the assigned coefficients in principle. These modifications will be carried out again using a bottom-up design, so that the certainty returned by the fuzzy engine is at least of 0.7 for the correct decision in the lowest level, and of 0.5 for other levels. Likewise, in order to limit the number of possible answers, incorrect question indexes may be modified making them lower.

This method has been proven considering the most frequent questions asked in the portal web administrator as our set of knowledge, which consists on a set of 117 questions. The elements to be found are the questions themselves and possibly related questions.

Test results for standard question recognition fit into five categories:

1.-The correct question is the only one found or the one that has the highest degree of certainty.

2.-The correct question is between the two with the highest certainty or is the one that has the second highest degree of certainty.

3.-The correct question is among the three with the highest degree of certainty or is the one that has the third highest certainty.

4.-The correct question is found but not among the three with the highest degree of certainty

5.-The correct question is not found.

4.2 Rule Definition

As mentioned above, rule definition corresponds to the administrator. Logically, the more inputs the engine has, the more rules there are. For example, for a three input engine, the inputs can take three values: LOW, MEDIUM and HIGH. The outputs can take the values of LOW, MEDIUM-LOW, MEDIUM-HIGH and HIGH. The inference rules defined are:

If all inputs are LOW, output is LOW.
If one input is MEDIUM and the others are LOW, output is MEDIUM-LOW.
If two inputs are MEDIUM and the others are LOW, output is MEDIUM-HIGH.
If all the input are MEDIUM or one input is HIGH, output is HIGH.

The possible combinations generate 27 rules for the fuzzy engine. A five input engine generates 243 rules.

4.3 Input Definition

For every question, 3 to 5 key words are defined. Nevertheless, the user may include in his consultation anywhere from 1 to 5 of these words. Defining an engine with only a few inputs causes rapid saturation of the system. This is a great handicap for precision: 90 % of correct answers are detected but only half of them are the first

option as may be seen in Table 1. Defining a five input engine produces values with a very low degree of certainty. Precision grows to 55 % but recall falls.

A solution to this problem is the use of variable thresholds. When all eventual outputs are below the fixed threshold, this threshold goes down until an output is found. If thresholds are variable, the system is more flexible and results are better.

Finally, the solution provided is to implement a flexible system with variable inputs. If the user consultation has at the most three key words, a three input fuzzy engine is used, whereas if the user consultation includes four or more key words, the system will use a five input fuzzy engine. Results are much better in terms of recall and precision. The user obtains the correct answer 97.75 % of the times and the first option 77.45 % of the times.

Table 1. Test results

Type of system / Test result category	First answer	Among the first two answers	Among the first three answers	Out of the first three answers	Failed answer
3 input engine	45 %	24 %	9 %	12 %	10 %
5 input system with fixed thresholds	55 %	12 %	3 %	1 %	29 %
5 input system with variable thresholds	70 %	14 %	3 %	1 %	12 %
Variable input system with fixed thresholds	77 %	16 %	4.5 %	1 %	1.5 %

5 Conclusions

A method of consultation based on a fuzzy logic application provided with an interface that one may interact with in natural language has been presented. The system takes advantage of converting any kind of object in a text object; this allows the application of text retrieval techniques. The other advantage of the method arises from fuzzy logic flexibility, which makes it possible to have a non-rigid term weighting in the stage of classification of the contents in the set of knowledge.

A method for IE in a set of knowledge is also proposed. We also present a modification of the classical VSM to define term weights, which are used as inputs to a fuzzy logic based system.

An eventual application of this system is the implementation of an intelligent agent to manage the information contained in an internet portal. So far the results obtained are good enough, as the number of correctly detected consultations is high.

References

1. Kwok, K.L.: A neural network for probabilistic information retrieval. In: Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval. Cambridge, Massachusetts, United States (1989)
2. Aronson, A.R., Rindflesch, T.C., Browne, A.C.: Exploiting a large thesaurus for information retrieval. In: Proceedings of RIAO, pp. 197–216 (1994)

3. Liu, S., Dong, M., Zhang, H., Li, R., Shi, Z.: An approach of multi-hierarchy text classification 2001. In: International Conferences on Info-tech and Info-net Proceedings. ICII 2001 - Beijing. vol. 3, Oct/ Nov. 2001 p. 95–100 (2001)
4. Lu, M., Hu, K., Wu, Y., Lu, Y., Zhou, L.: SECTCS: towards improving VSM and Naive Bayesian classifier. In: 2002 IEEE International Conference on Systems, Man and Cybernetics.vol. 5, p. 5 (October 2002)
5. Pal, S.K., Talwar, V., Mitra, P.: Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions. In: IEEE Transactions on Neural Networks Vol. 13(5), pp 1163–1177 (September 2002)
6. Rios, S.A., Velásquez, J.D., Yasuda, H., Aoki, T.: Conceptual Classification to Improve a Web Site Content. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) IDEAL 2006. LNCS, vol. 4224, pp. 869–877. Springer, Heidelberg (2006)
7. Chakrabarti, S.: Data Mining for hypertext: a tutorial survey. ACM SIGKDD Explorations, Newsletter of the Special Interest Group on Knowledge Discovery and Data Mining (2000)
8. Papadakis, N.K., Skoutas, D., Raftopoulos, K., Varvarigou, T.A.: STAVIES: A System for Information Extraction from Unknown Web Data Sources through Automatic Web Wrapper Generation Using Clustering Techniques. IEEE Transactions on Knowledge and Data. Engineering 17(12), 1638–1652 (2005)
9. Liu, L., Buttler, D., Caverlee, J., Pu, C., Zhang, J.: A methodical approach to extracting interesting objects from dynamic web pages. Int. Journal of Web and Grid Services 1(2), 165–195 (2005)
10. Salton, G.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
11. van Rijsbergen, C.J.: Information retrieval. Butterworths (1979)
12. Salton, G., Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. Technical Report TR87-881, Department of Computer Science, Cornell University, 1987. Information Processing and Management 32(4), 431–443 (1996)
13. Salton, G.: Automatic Text Processing. Addison-Wesley, Reading (1988)
14. Lee, D.L, Chuang, H., Seamons, K.: Document ranking and the vector-space model. Software, IEEE 14(2), 67–75 (1997)
15. Zhao, Y., Karypis, G.: Improving precategorized collection retrieval by using supervised term weighting schemes. In: Proceedings of the International Conference on Information Technology: Coding and Computing, 8-10 April 2002, pp. 16–21 (2002)
16. Martín del Brío, B., Sanz Molina, A.: Redes neuronales y sistemas borrosos Ra-Ma (2001)
17. Bouaziz, T., Wolski, A.: Applying Fuzzy Events to Approximate Reasoning in Active Databases. In: Proc. Sixth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'97), July 1-5, Barcelona, Spain (1997)
18. Ruiz, M.E., Srinivasan, P.: Automatic Text Categorization Using Neural Networks. Advances in Classification Research. In: Proceedings of the 8th ASIS SIG/CR Classification Research Workshop. Efthimiadis, E. (ed.) Information Today, Medford:New Jersey. vol. 8, pp 59–72 (1998)

Efficient Methods in Finding Aggregate Nearest Neighbor by Projection-Based Filtering

Yanmin Luo[1,2], Hanxiong Chen[1], Kazutaka Furuse[1], and Nobuo Ohbo[1]

[1] Dept.Computer Science,University of Tsukuba,Tennoudai 1-1-1,Tsukuba,Ibaraki 305,Japan
{lym,chx,furuse,ohbo}@dblab.is.tsukuba.ac.jp
[2] Dept. Computer Science, HuaQiao University, QuanZhou, FuJian, China, 362021
lym@hqu.edu.cn

Abstract. Aggregate Nearest Neighbor (ANN) queries developing from Nearest Neighbor (NN) queries are the relatively new query type in spatial database and data mining. An ANN queries return the object that minimizes an aggregate distance function with respect to a set of query points. Because of the multiple query points, ANN queries are much more complex than NN queries. For optimizing the query processing and improving the query efficiency, many ANN queries algorithms utilizes pruning strategies. In this paper, we propose two points projecting based ANN queries algorithms which can efficiently prune the data points without indexing. We project the query points into special "line", on which we analyses their distributing, then pruning the search space. Unlike many other algorithms based on the data index mechanisms, our algorithms avoid the curse of dimensionality and are effective and efficient in both high dimensional space and metric space. We conduct experimental studies using both real dataset and synthetic datasets to compare and evaluate their efficiencies.

Keywords: Aggregate Nearest Neighbor, Pruning strategies, Search Region.

1 Introduction

Similarity search is a very important query operation in data mining especial for information retrieval. It is most common that the similarity between two retrieval objects can be measured by their distance metric. Nearest Neighbor queries can be used to compute the similarity between objects in large database especially in spatial database retrieval such as in Geographic Information System(GIS). For minimizing the query processing time in similarity queries, most of the NN queries algorithms use the index mechanisms in their query processing. Utilizing the index-structure helps pruning or restriction of the search space. In spatial databases, datasets are usually indexed by spatial access methods (SAM) such as the R-tree[3] and the R*-tree[16]. Therefore many NN queries algorithms used them to implement query in multi-dimensional data space[1,9,15]. Some metric indices such as MB$^+$-tree[14], GNAT[17] and MVP-tree[18] also have been proposed for similarity queries.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 821–833, 2007.
© Springer-Verlag Berlin Heidelberg 2007

ANN Queries are novel forms of NN queries. Given two sets of points in spatial database P and Q, $P=\{p_1,p_2,\ldots p_N\}$ is the static source dataset and $Q=\{q_1,q_2,q_3 \ldots\ldots,q_n\}$ is the query points set. As defined in [6], the aggregate distance between a data point and query points set Q can be expressed by $adist(p,Q)=f(|pq_1|, |pq_2|,\ldots,|pq_n|)$, where $|pq_i|$ is the Euclidean distance between point p and qi. Given a set P of static points, ANN Queries retrieves the data point p in P having minimal aggregate distance to Q. For ANN, different function f gives ANN different meaning. For example, if f is a *sum* function, the ANN queries will find the point p with minimal total aggregate distance to Q. And if f is the max function, the ANN query will return the point p which minimizes the maximum distance to the points in Q. Fig.1 shows the example. Fig.1(a) shows the case that f is the sum function, and the ANN queries will report the data point p_5 because it minimizes the total distance to Q. In other words, $adist(p_5,Q)=\Sigma_{i=1..3} |p_5q_i|<=adist(p,Q)$ for all p in P. Fig.1(b) shows the case when f is a *max* function, where the ANN queries report the data point p_7 which minimizes the maximum distance to Q. This is to say that $adist(p_7,Q)=max_{i=1..3}|p_7q_i|=|p_7q_1|<= adist(p,Q)$ for all p in P. Considering that several persons at specific locations (query points of Q) want to meet at a certain restaurant (a data point of P). The sum aggregate query return the restaurant (p_5) which leads to the minimum sum of distances that all the persons have to move while the max aggregate query return the restaurant (p_7) where all members can arrive together in the shortest time. Furthermore, ANN query now is becoming more and more important in many domains, such as clustering [2], outlier detection[4], GIS and mobile computing application [5,10,12].

Now NN problems and ANN problems are reasonably well solved for low dimensional applications for which many efficient indexing structures have been proposed. A wide variety of indexes have been proposed such as R-tree, most of them work well in low dimensional space. Many studies have shown that traditional indexing methods fail in high dimensional space. According to [11], NN and RNN search could be meaningless in high dimensional spaces due to the well-known curse of dimensionality. This motivates our interesting in non-indexing approaches. In order to compute as few distances as possible, some pruning strategies are always used to optimize the query processing in ANN queries [6,7,8]. Given a static dataset and several query points, and suppose that the static dataset is indexed by an R-tree. [6,7]

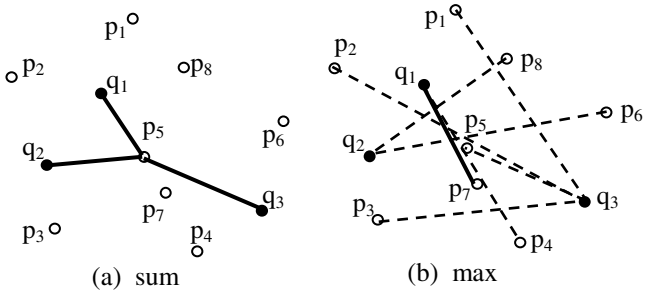


Fig. 1. Examples of ANN Queries. The solid points are query points and the hollow points are the data points. The point(s) connected by bold line(s) are the ANN results.

develop several ANN query algorithms for processing both memory-resident and disk-resident query sets. Some metrics are utilized to prune the search space for queries optimization. [8] uses an ellipse to approximate the extent of all query points, and the distance or MBR derived from the ellipse is used to prune intermediate index nodes during search via the R*-tree.

In this paper, we propose two non-indexing pruning strategies for ANN query processing. We assume that all query points can fit in main memory and only consider the *sum* function. Our methods find the search region in data space and prune other region. The remainder of this paper is organized as follows. Section 2 gives a brief survey of related work. Section 3 describes our pruning strategies for ANN queries and gives out the algorithms. Section 4 experimentally evaluates the effectiveness of our algorithms, and Section 5 concludes the paper.

2 Related Works

Similarity search in metric space is performed by measuring the (dis)similarity between objects by their distances. We adopted similar notations as those used in [13] with application-specific metric distance function d. Formally a metric space is a pair, $MS = (O, d)$, where O is a set of data values and d is a total (distance) function with the following properties (the metric postulate):

$$x, y \in O; \quad d(x, y) = d(y, x). \tag{1}$$

$$x \in O; \quad d(x, x) = 0. \tag{2}$$

$$x, y \in O; \quad x \neq y \rightarrow 0 < d(x, y) < \infty. \tag{3}$$

$$x, y, z \in O; \quad d(x, z) \leq d(x, y) + d(y, z). \tag{4}$$

Given a query object $q \in O$ and a nonnegative radius r, a r-neighbor search (r-N search) of q is to retrieve the objects in the r-neighbor of q:$\mathrm{RN}(q,r)=\{o | o \in O \wedge d(q,o) \leq r\}$. We call this region by r-neighbor region, see Fig. 2(a).

Assuming that the set Q of query points fits in memory and the data points are indexed by an R-tree, Dimitris Papadias et al. introduced the algorithms for memory -resident ANN queries in [7], one of them is Multiple Query Method (MQM). MQM utilizes the main idea of the threshold algorithm. It executes incremental NN search for each point in Q, and combines their results. MQM stores a threshold t_i for each query point q_i, The sum of all thresholds is defined to be the total threshold T. MQM computes the nearest neighbor for each query point incrementally, updates the thresholds and gets the current best nearest neighbor with the best distance, until threshold T is larger than best distance. In order to achieve locality of the node accesses for individual queries, MQM sorts the points in Q according to their Hilbert value. Thus, two subsequent queries are likely to correspond to nearby points and access similar R-tree nodes. But for different query points Q, MQM would have to sort again. The algorithm for computing nearest neighbors of query points should be incremental because the termination condition is not known in advance.

3 Our Methods

In this section we describe our methods on ANN queries and give out the algorithms. For the following discussion, we consider Euclidean distance and 2-Dimendional point datasets. But the proposed techniques are applicable to higher dimension. And we only focus on the sum function due to its significance in practice. Table 1 lists the symbols and their definitions in our algorithms.

Table 1. Symbols in this paper

Symbols	Definitions
N	Number of objects (points) in database
n	Number of objects in query dataset
$Q=\{q_1,q_2 \ldots q_n\}$	The query dataset
MBR	Minimum Bounding Rectangle
M	The area of MBR
$d(p,q)$	Distance function between two objects
$D(p,q)=\|pq\|$	Euclidean distance between two objects
$adist(p,Q)$	$=f(\|pq_1\|,\|pq_2\|,\ldots,\|pq_n\|)$, aggregate distance
$C(q,r)$	$=\{p \mid p\in M, \|pq\|<=r\}$
$A(q,r_1,r_2)$	$=\{ p \mid p\in M,\ r_1<=\|pq\|<=r_2 \}=\{C(q,r_2)-C(q,r_1)\ \}$
$S(p)$	$\equiv S(p,Q)=\bigcup C(q_i,\|pq_i\|)$

Thinking about the points in dataset P are distributing uniformly in data space. As for the *sum* function, it is seen from Fig.2(b) that the result of ANN queries (we call it the best ANN point) should be in the MBR of the query points set. The points outside of the MBR can be pruned. We discuss non-index pruning methods. Pruning means we have to find a search region in data space and get the best ANN point in this region, instead of searching in full data space. The most important technique of our methods is how to find the search region which is equal to or includes the MBR of query points set and can substitute it. As we introduce our algorithms, we always first analyze how to get the proper search region.

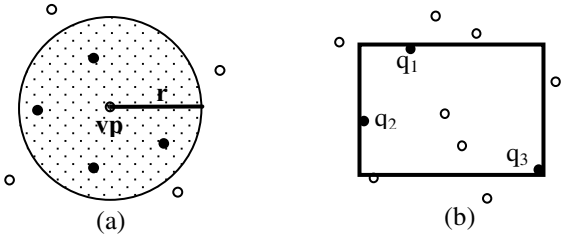


Fig. 2. (a) an example of the r-neighbor region of pivot vp , (b) an example of the MBR of Q, which includes three points q_1,q_2,q_3

3.1 *vp* -ANN Algorithm

Given a point *vp* and radius *r* we can divide the data space into two partitions. The shadow region in Fig.2 (a) is the r-neighbor region of *vp*. We call the other region by *out-neighbor* region. Assuming that *p* is the best ANN point, then *p* minimizes $adist(p,Q) = sum(|pq_1|,|pq_2|,\ldots,|pq_n|)$. It can be ensured that for every point q_i the best ANN point should be in its r-neighbor region. This means that the out-neighbor region of q_i can be pruned in our ANN queries. According to table 1, we can describe the r-neighbor region of the point *vp* by $C(vp,r) = \{p|p \in P, |p,vp| <= r\}$. For each point of q_i in *Q* we can get the search region of ANN query by the definition of Lemma 1.

Lemma 1. *Let* $R_i = C(q_i,r_i)$ *be r-neighbor region of* q_i, *and* $R = \bigcup_{i=1..n} R_i$ *be the union area of* R_i. *If each* r_i *is chosen that* $\bigcap_{i=1..n} R_i$ *is not empty, then the best ANN point must be found in R.*

Lemma 1 is clear and by it, we can search only region *R* to find the best ANN point. Now what we have to do is to select r_i for each query point q_i. We have to choose each r_i carefully in order to make the initial region *R* as small as possible. One way to do this is to choose a point *vp* in dataset *P* randomly, and select the Euclidean distance between *vp* and q_i as r_i. However, a randomly selected *vp* point may make *R* cover almost all of the data points in dataset *P*, and only a few points can be pruned. Therefore a good *vp* would lead to fewer distance calculations. Thinking over all the query points we know the aggregate centroid of *Q* is the best choice of *vp*, because as for sum function this point is the one in data space that minimizes the value of $adist(q,Q)$. We can say that the aggregate nearest neighbor is a point of *P* "near" the aggregate centroid of *Q*. The aggregate centroid depends on the function f and its exact computation is not always possible. It should be computed at the initialization phase of *vp*-ANN. Thinking about the case *f=sum*, *q* minimizes the function $adist(q,Q)=\sum_{i=1..n}|qq_i|$. Enlightened from the SPM algorithm in [6], let (*x*,*y*) be the coordinates of q and (x_i,y_i) be the coordinates of query point q_i . Since the partial derivatives of function $adist(q,Q)$ with respect to its independent variables x and y are zero at the centroid q, we have the following equations:

$$\partial adist(q,Q)/\partial x = \sum_{i=1..n} ((x-x_i)/sqrt((x-x_i)^2+(y-y_i)^2))=0. \quad (5)$$

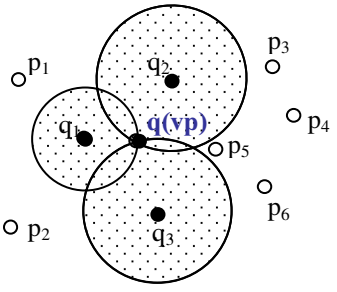
$$\partial adist(q,Q)/\partial y = \sum_{i=1..n} ((y-y_i)/sqrt((x-x_i)^2+(y-y_i)^2))=0. \quad (6)$$

Unfortunately, the above equations cannot be solved into closed form for $n>2$. This is to say that it is difficult to calculate this centroid. In our implementation, we just think about the case of coordinate space and use the geometric centroid to substitute it. The geometric centroid q(x,y) of *Q* is:

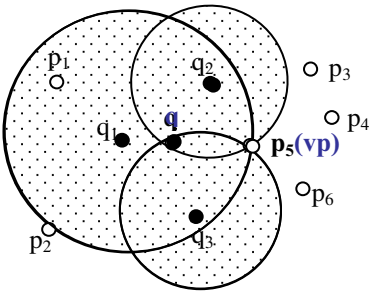
$$x = (1/n)\cdot\sum_{i=1..n} x_i. \quad (7)$$

$$y = (1/n)\cdot\sum_{i=1..n} y_i. \tag{8}$$

Although the geometric centroid of Q is the best choice of *vp*, it is not certain a point of the dataset P. If the dimension is high and the data points are few and far away from each other in data space, it is possible that R would be a NULL set, see Fig.3(a). For this reason we find the nearest neighbor of the geometric centriod in P (Assuming it is p_k), and choose it(p_k) as the *vp* point instead of the geometric centriod. On this *vp*, there is at least one point p_k in R, Fig.3(b) shows this. p_5 is selected to be *vp*, for it is the nearest neighbor of q. Fig.4 shows *vp*-ANN algorithm.



(a) *vp* is the geometric centroid q of Q



(b) *vp* is the nearest neighbor of *q*

Fig. 3. The search region according to different *vp*. *q* is the geometric centroid of Q but q is not a point of P. p_5 is the nearest neighbor of q of P. (a) selects q as the pivot, and (b) selects p_5 as pivot.

In the *vp*-ANN algorithm the dataset R always covers the region of the MBR of query points set, therefore this algorithm is strong and always can get the best ANN query point. We will discuss its efficiency in section 4.

Algorithm *vp*-ANN(Q:set of the query points) /* *vp* is the vantage point */
1. initialize $R=\Phi$
2. compute the aggregate centroid *q* of Q
3. find *vp* point as the nearest neighbor of q in dataset P
4. for each point q_i in query dataset Q do
5. compute r_i; // $r_i=|q_i vp|$
6. $R_i = C(q_i, r_i)$;
7. Updata R; // $R=R \cup R_i$
8. end for
9. search R to find the best ANN point;

Fig. 4. The *vp*-ANN algorithm

3.2 Projection-Based Pruning Algorithm

In this section, if a search region is small enough, and can cover the search region discussed in *vp*-ANN algorithm, then we say that this search region is a "*satisfied*" one. The heart of the method proposed in this section is to project query points into a carefully selected 'line'. The result of projection can approximately reflect the distributing of query points. On that we can determine the search region in which the query points distribute concentrated, and prune other region of data space. To do this, firstly we choose two so-called *pivots* points: p_a and p_b and consider the 'line' that passes through them in data space. The algorithm to choose pivot points will be discussed later in this section (see Fig.8). Then we project every query point into this line. The distance between pivot p_a and the projecting point of the query point on that line can be computed by using the cosine law. We mark this distance by $proj(p_a,q)$ and illustrate it in Fig.5. According to the cosine law, we can get the Eq.9, where $D(p_a,q)$ is the Euclidean distance between p_a and q. By this equation we know $Proj(p_a,q)$ is great than 0 when $\alpha<\pi/2$ and less then 0 when $\alpha>\pi/2$. This means that $Proj(p_a,q)\neq D(p_a,q')$, but $|Proj(p_a,q)|=D(p_a,q')$.

$$Proj(p_a,q)=(D(p_a,q)^2+D(p_a,p_b)^2-D(p_b,q)^2)/(2D(p_a,p_b)). \quad (9)$$

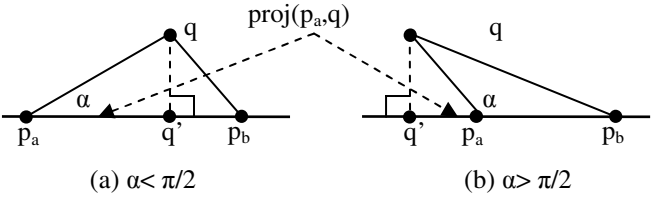
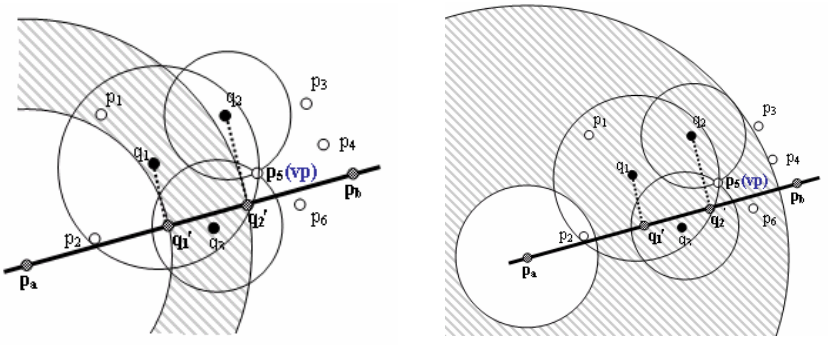


Fig. 5. Illustration of the projecting law

For each query point q_i we repeat the projecting and get the value of $proj(p_a,q_i)$. Then we find out the query point q_{max} with the maximal value of $proj(p_a,q_{max})$, and the query point q_{min} with minimal value of $proj(p_a,q_{min})$. Let q_{max}' be the projecting point of q_{max} on the line and q_{min}' be the projecting point of q_{min} on the line. Let $r_{max}=proj(p_a,q_{max})$ and $r_{min}=proj(p_a,q_{min})$. Also if $proj(p_a,q_{min})<0$, let $r_{min}=0$. Taking p_a as the centre of a circle we draw two circles with the radii r_{max} and r_{min} respectively, see Fig.6 (a). There must be a cirque between the two circles, see the shadow region. We call this region the candidate region, and express it by $A(p_a,r_{min},r_{max})$ (before amending) as in Tab.1. Comparing this candidate region with the search region in *vp*-ANN algorithm, the candidate region cannot cover all or most of the search region in *vp*-ANN algorithm, and even the point q_{max} (see q_2 in Fig.6 (a)) is not be included. Therefore this region is not a satisfied one and cannot be the search region. We have to extend this cirque region (candidate region), and enlarge it to cover all or most of the search region in *vp*-ANN algorithm. In our implementation we modify the value of r_{max} and r_{min} according to Eq.10 and Eq.11. Now we redraw the two circles and get the new cirque $A(p_a,r_{min},r_{max})$ which is the shadow region in Fig.6(b). We can see that the new cirque covers all or most of the search region of the *vp*-ANN algorithm.

$$r_{max} = r_{max} + D(q_{max}, q_{max}'). \tag{10}$$

$$r_{min} = r_{min} - D(q_{min}, q_{min}') \ \ (if\ r_{min} > 0). \tag{11}$$



(a) $A(p_a, r_{min}, r_{max})$ before amending (b) $A(p_a, r_{min}, r_{max})$ after amending

Fig. 6. The search region according to different *vp* points

Unfortunately this candidate region is still too large. As we can see in Fig.6 (b), the candidate region may include almost all the points of dataset and only a small number of points can be pruned. But this candidate region will be used to get the search region in our algorithm. Giving two cirques with different centres, if they are close to each other, there will be a overlapping region between them. Our strategy is: firstly, select two points p_{a1} and p_{b1}, and draw the line passing through them. Using the theory above, we can get the 1st candidate region: $A(p_{a1}, r_{min1}, r_{max1})$. Then we select the other two points p_{a2} and p_{b2} to determine the second line which different from the first one, and get the second candidate region $A(p_{a2}, r_{min2}, r_{max2})$. Because their centres: p_{a1} and p_{a2} are different points, and we project the same dataset P to the two different line, there must have an overlapping region between $A(p_{a1}, r_{min1}, r_{max1})$ and $A(p_{a2}, r_{min2}, r_{max2})$. Let $S = A(p_{a1}, r_{min1}, r_{max1}) \cap A(p_{a2}, r_{min2}, r_{max2})$, see Fig.7. We select S as the search region in our algorithm, because it is a "satisfied" one.

From the discussion above, it can be seen that pivot points affect the performance significantly. Clearly, we would find a line on which not only the projections of the query points are as far apart from each other as possible but also let all or most of the values of $proj(p_a, q_i)$ be positive. To achieve this, the pivot points p_a and p_b should maximize the distance $|p_a p_b|$. However, this spends $O(N^2)$ distance computation. Moreover, if the line passes through the center of query set Q, the distance $|q_{max} q_{max}'|$ in Eq.6 and Eq.7 will be as small as possible. That would make the area $A(p_a, r_{min}, r_{max})$ smaller too. This means we can prune more points of dataset P. For those reason, we propose a linear heuristic algorithm to get the pivot points which requires only $O(N)$ distance computations. Firstly we compute the aggregate centroid Q which is also necessarily in projecting operation, and compute its farthest point as the 1st pivot (p_a). Then we find the farthest point apart from p_a in dataset P, and make it be the 2nd pivot (p_b). The pivot points choosing algorithm is showed in Fig.8.

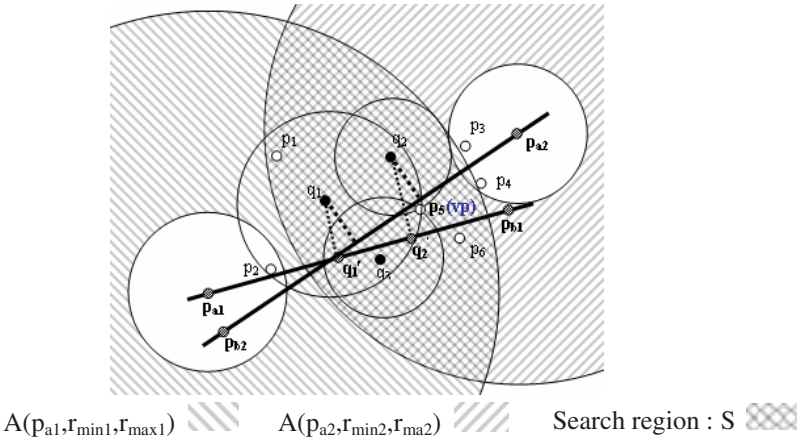


Fig. 7. The search region of projection-based algorithm

Algorithm pivot_points_choosing(Q:set of the query points)
1. let q_{cen} = the aggregate centroid of Q
2. let $p_a = arg\ max_{\ p}\ \{|pq_{cen}|\}$ // *the farthest point from* q_{cen}
3. let $p_b = arg\ max_{\ p}\ \{|pp_a|\}$
4. return p_a and p_b

Fig. 8. The pivot point choosing algorithm of projection-based algorithm

Algorithm projection_based_pruning algorithm(Q:set of the query points)
1. initialize S=Φ;
2. for s=1, 2 do
3. $\{p_{as}, p_{bs}\}$= **pivot_points_choosing**(Q)
4. for each point q_i in Q do
   project q_i to the line $p_{as}p_{bs}$ passing through p_{as} and p_{bs}
   calculate the value of $proj(p_{as},q_i)$
   endfor
5. get the points $p_{max,s}$,$p_{min,s}$ with the maximal and minimal projection in line $p_{as}p_{bs}$
6. calculate the value of $r_{max,s}$ and $r_{min,s}$
7. pointset_As= $A(p_{as},r_{min,s},r_{max,s})$ /*get the candidate region */
8. endfor
9. S = *pointset_A1* ∩ *pointset_A2* /*get the search region */
10. retrieve in S and get the best ANN point;

Fig. 9. The projection-based pruning algorithm

Now we are ready to describe the projection-based pruning algorithm. Our algorithm accepts the query points set as input and find out the best ANN point in dataset. Fig.9 shows the pseudo code of the projection-based pruning algorithm.

4 Experiments

All the experiments explained in this section were programmed in C++ under the GNU/Linux operating system, and are run on an Intel Pentium(R) 4 CPU 2.40GHz PC, 512MB main memory. Datasets are as the following, with each coordinate been normalized to [0, 1]:

(i) WUpppoint, points of populated places of West part of United Stated of America (10493) which are available at www.rtreeportal.org.

(ii) A synthetic high dimensional dataset generated randomly according to the uniform distribution.

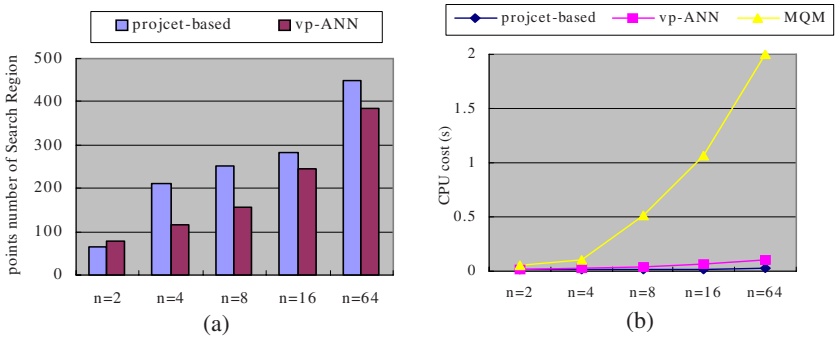


Fig. 10. Comparison on query set size (N=10000, M=0.08, 2-Dimension)

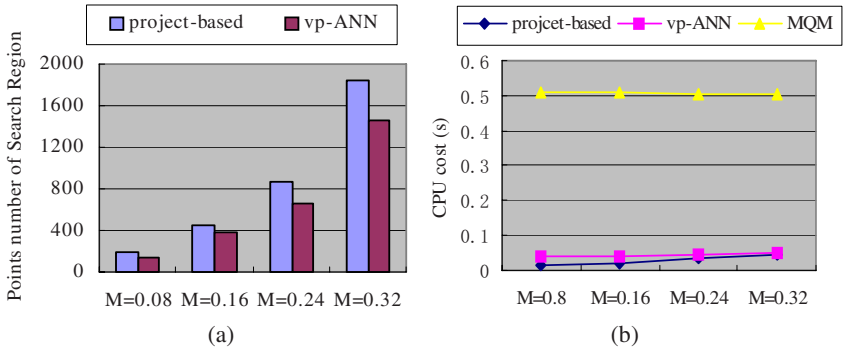


Fig. 11. Comparison on MBR area (N=10000, n=8, 2-Dimension)

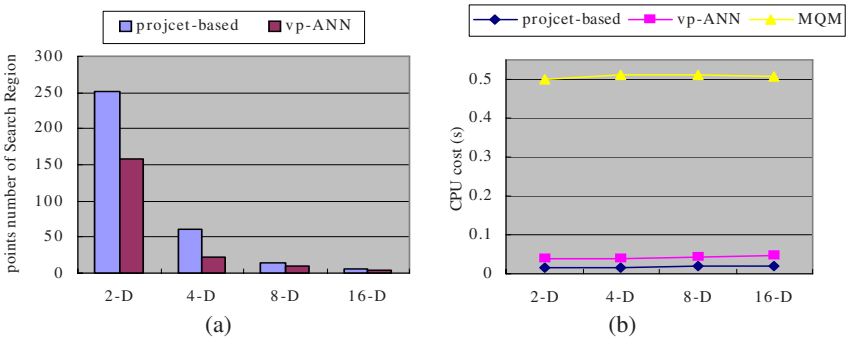


Fig. 12. Comparison on dimensionality (N=10000, n=8, M=0.08)

In our experiments, the query dataset is memory-resident. We use workloads of 50 queries. For each query, n is the point number in query points set Q. The query points are generated randomly in the workspace of P, and distributed uniformly in a MBR of area M. We compare the CPU cost and the screening effect (number of data points in search region) of our algorithms against MQM algorithm, studying the following cases of variation. That is, number of dimensions of data space, query set size (number of query points in Q) and the area of the MBR of Q. Fig.10 shows the average number of points in search region and CPU cost as functions of n on datasets (i). Fig.10 (a) shows that with the increasing of n, the point number of search region will become more. The points number of search region for projection-based always a little bigger than *vp*-ANN. This is because the search region of projecting-based always covers that of *vp*-ANN. Nevertheless, the CPU cost of projection-based is low. Fig.10 (b) shows that a lager n makes the CPU costs of all the three algorithms increase. But the CPU cost of projection-based algorithm goes up slower than the others, and MQM becomes worse rapidly. As in Fig.11, with the increasing of M, both the points number in search region and CPU cost of projection-based and *vp*-ANN algorithms increase as well. And the CPU cost of projection-based one increases more sharply while the CPU cost of MQM is almost unchanged. n is fixative, therefore the value of n and the area M become the dominative factors of our algorithms. For CPU cost, the projection-based algorithm is better than *vp*-ANN algorithm. As in Fig.12, if the dimensionality increases, the data is distributed more infrequently, and the number of data points in the search region will decrease sharply. While the CPU cost is stable and increases slightly, for fixed n and M. This result shows that the non-indexing algorithms perform well in high dimension space.

Focusing on the CPU costs, we know if n is small the CPU cost of all the three algorithms are similar. If n increases, the CPU cost of *vp*-ANN algorithm increases faster than projection-based algorithm. This is because *vp*-ANN algorithm has to calculate more r-neighbor region of q_i to find candidates, while a larger n will not influence the calculations of the candidate regions in projection-based method. In Fig.10 (b) we can see that when n is larger, the CPU cost of MQM method increases

very sharply. This is because MQM has to sort the points of P for every query point q in Q according to their distance to the points in P. Thus a larger n will let the sorting become more time-consuming. As for different query, the query points set Q will change, and we have to redo the sorting in MQM algorithm. This is the main deficiency of MQM algorithm. On the other hand a different Q will not influence the executing of our algorithms.

5 Conclusion

ANN Queries are the extension of NN queries but are more complex because of multiple query points. Based on the coordinate space, we consider the case of *sum* function in ANN queries. Assuming that Q is memory-resident we describe and analyze two non-index algorithms for best ANN queries, and compare them with index-based methods. Our strategy finds the proper search region and enables pruning of irrelevant ones. The experimental results demonstrate that the methods we proposed performance well, especially in high dimensional space. In the future we intend to explore the case of other functions.

References

[1] Papadopoulos, A., Manolopoulos, Y.: Performance of Nearest Neighbor Queries in R-Trees. In: Afrati, F.N., Kolaitis, P.G. (eds.) ICDT 1997. LNCS, vol. 1186, pp. 394–408. Springer, Heidelberg (1996)
[2] Jain, A.K., Narasimha Murty, M., Flynn, P.J.: Data Clustering: A Review. ACM Comput. Surv. 31(3), 264–323 (1999)
[3] Guttman, A.: R-Trees: A Dynamic Index Structure for Spatial Searching. In: Proc. of ACM SIGMOD Int'l Conference, pp. 47–57 (1984)
[4] Aggarwal, C.C., Yu, P.S.: Outlier Detection for High Dimensional Data. In: SIGMOD Conference, pp. 37–46 (2001)
[5] Jensen, C.S., Kolárvr, J.: Torben Bach Pedersen, Igor Timko: Nearest neighbor queries in road networks. In: GIS 2003, pp. 1–8 (2003)
[6] Papadias, D., Tao, Y., Mouratidis, K., Hui, C.K.: Aggregate nearest neighbor queries in spatial databases. ACM Trans. Database Syst. 30(2), 529–576 (2005)
[7] Papadias, D., Shen, Q., Tao, Y., Mouratidis, K.: Group Nearest Neighbor Queries. In: ICDE 2004, pp. 301–312 (2004)
[8] Li, H., Lu, H., Huang, B., Huang, Z.: Two ellipse-based pruning methods for group nearest neighbor queries. In: GIS 2005, pp. 192–199 (2005)
[9] Uhlmann, J.K.: Satisfying general proximity/ similary queries with metric trees. Information Processing Letters 40, 175–179 (1991)
[10] Lin, K.-I., Nolen, M., Kommeneni, K.: Utilizing Indexes for Approximate and On-Line Nearest Neighbor Queries. In: IDEAS 2005, pp. 83–88 (2005)
[11] Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is Nearest Neighbors Meaningful? In: Proc. of the Int. Conf. Database Theorie, pp. 217–235 (1999)
[12] Yiu, M.L., Mamoulis, N., Papadias, D.: Aggregate Nearest Neighbor Queries in Road Networks. IEEE Trans. Knowl. Data Eng. 17(6), 820–833 (2005)

[13] Paolo Ciaccia, M.P., Zezula, P.: M-tree: An efficient access method for similarity search in metric spaces. In: Proceedings of the 23rd VLDB Conference, Athenes, Greece, 1997, pp. 357–368 (1997)
[14] Ishikawa, M., Chen, H., Furuse, K., Yu, J.X., Ohbo, N.: MB+Tree: A Dynamically Updatable Metric Index for Similarity Searches. Web.-Age. Information Management, 356–373 (2000)
[15] Roussopoulos, N., Kelley, S., Vincent, F.: Nearest Neighbor Queries. In: SIGMOD Conference 1995, pp. 71–79 (1995)
[16] Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B.: The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. In: Proc. of ACM SIGMOD Int'l Conference, pp. 322–331 (1990)
[17] Brin, S.: Near neighbor search in large metric spaces. In: Proceedings of the 21st VLDB Conference, pp. 574–584 (1995)
[18] Bozkaya, T., Ozsoyoglu, M.: Distance-based inde- xing for high-dimensional metric spaces. In: Proceedings of the ACM SIGMOD Conference on Management of Data, Tucson, Arizona, pp. 357–368 (May 1997)

On Multicast Routing Based on Route Optimization in Network Mobility

Jong-Ki Kim[1], Kisoeb Park[2], and Moonseong Kim[2,⋆]

[1] Department of Information Security
Konyang University, Korea
jkkim@konyang.ac.kr
[2] Sungkyunkwan University, Korea
{kisoeb, moonseong}@skku.edu

Abstract. Mobile IP (MIP) is designed for supporting mobile nodes. However, the technology does not sufficiently handle **NE**twork **MO**bility (NEMO). The NEMO Basic Support (NBS) [1] ensures session continuity for all the nodes in a MObile NETwork (MONET). Since the protocol is based on MIP, when used to support multicast for NEMO, it inherits the same fundamental problems as MIP, such as tunnel convergence. Recently, many Route Optimization (RO) schemes have been proposed for the pinball routing problem in NEMO. And we should use the RO schemes for multicasting in NEMO. However, the tunnel convergence problem may be remained as unsolvable problem. Hence, we propose the similar source-based routing multicast scheme. In this paper, we present the multicast scheme based on well-known RO scheme Path Control Header (PCH) [2] in NEMO environments. Furthermore, the proposed scheme's recovery time is more efficient than other well-known multicast schemes.

Keywords: Multicasting in NEMO, Network Mobility (NEMO), Path Control Header (PCH) Protocol.

1 Introduction

The goal of traditional mobility support is to provide continuous Internet connectivity to mobile hosts. Mobile IPv4 (MIPv4) [3] and Mobile IPv6 (MIPv6) [4] have introduced mobility support for IPv4 and IPv6 nodes, respectively. In contrast, **NE**twork **MO**bility (NEMO) is concerned with situations where an entire network (composed by one or more subnets) dynamically changes its point of attachment to the Internet and thus its accessibility in the topology. NEMO Basic Support (NBS) [1] ensures session continuity for all the nodes in a MObile NETwork (MONET), even when the Mobile Router (MR) changes its point of attachment to the Internet. It also provides connectivity and reachability for all nodes in the MONET as it moves. The protocol relies on the establishment of a bi-directional tunnel between the MR and its Home Agent (HA).

⋆ Corresponding author.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 834–843, 2007.
© Springer-Verlag Berlin Heidelberg 2007

While mobility support for IPv6 unicast sessions is fully addressed by the mechanism mentioned above, it does not discuss the case of IPv6 multicast [5]. The issue is whether to enable session continuity both for multicast receivers and for multicast sources located inside a MONET, while an MR is moving. Since multicast routing differs greatly from unicast routing, the HA forwarding mechanism defined by NBS for unicast packets, does not serve the forwarding of multicast packets. The key reason is that multicast packets are sent to multicast receivers with multicast addresses that are unrelated to the prefixes of the networks where potential multicast receivers are located (*e.g.*, the MONET prefix). A possible solution is the use of the MR-HA bi-directional tunnel for the forwarding of multicast traffic as well, in both directions, between the MR and its home link.

In early MIP technologies, basic architectures such as Bi-directional Tunneling (BT) [6] and Remote Subscription (RS) were suggested to provide multicast service for mobile hosts. Because the NBS architecture is based on MIP, it inherits from MIP the same fundamental problems such as tunnel convergence problem, and mass of multicast problem, when it is used to support multicast for MONETs. Therefore, multicast support mechanisms suitable for NEMO environments must be investigated. Recently, many Route Optimization (RO) schemes have been proposed for the pinball routing problem in NEMO. And we should use the RO schemes for multicasting in NEMO. However, the tunnel convergence problem may be remained as unsolvable problem. In this paper, hence, we propose the similar source based routing multicast scheme based on well-known RO scheme, Path Control Header (PCH) [2], in NEMO environments.

The rest of the paper is organized as follows. Section 2 presents the previous works which are to compare with our protocols and introduces RO scheme PCH. Section 3 describes the proposed scheme, and Section 4 presents analytical models to evaluate the proposed scheme. Section 5 concludes this paper.

2 Previous Works

2.1 Two Basic Multicast Protocols Based on Mobile IP

Subscription on a foreign network is the simplest option for obtaining multicast services since it does not contain special encapsulation requirements, and only operates using existing protocols. With this option, the MN is required to resubscribe to the multicast group on each foreign network, and must use a co-located Care of Address (CoA). If the MN is highly mobile, however, packets will be lost owing to the set-up time associated with multicast subscription, and therefore, in this case this method is not preferred. If QoS requirements are crucial, or a host is likely to be stationary for an extended period of time, then this option is preferred, particularly if the MN can unsubscribe prior to leaving. Remote Subscription (RS) does provide the most efficient delivery of multicast datagrams, but this service may come at a high price for the networks involved and the multicast routers that must manage the multicast tree.

Bi-directional Tunneling (BT) for unicast datagrams is discussed in the standard [6]. This method is designed to solve the problem of topologically incorrect source addresses in datagrams by requiring traffic from the MN to be routed back to the home network through a Foreign Agent (FA) to the HA tunnel. With bi-directional tunneled multicast, the mobile routing agent on the home network must also be a multicast router. Using this option, subscriptions are done through the HA. When the MN is away from home, a bi-directional tunnel to the HA is set up. This permits both sending and receiving of multicast datagrams, with the same delivery guarantees given to fixed hosts. One disadvantage of this situation, is that if multiple MNs on the same foreign network belong to the same multicast group then duplicate copies of the multicast packets will arrive at the foreign network. This problem negates some of the advantages of using multicast in the first place.

2.2 NEMO Basic Support Protocol

In order to support NEMO, various basic requirements on existing networks must be satisfied [1]. The majority of the requirements are met by employing an MR, which controls network mobility. This MR provides a path to on-link nodes in its network and guarantees network connectivity to the Internet, because the MR not only acts as a MN, but also as a router. To support these basic requirements in the NEMO Basic Support (NBS) architecture, bi-directional tunneling is necessary between the MR and its HA so that on-link nodes in the MR's network can be reachable in the Internet. A Correspondent Node (CN) may not know where the target node is placed initially during communications after a MONET finishes its movement. Therefore, the HA must retrieve the location of the node and forward data to it. The location information of the node that the CN desires to communicate with, however, does not exist in the binding list of the HA. This is because the node did not initiate any registration request message for movement notification. Nevertheless, the HA can find out the location information of the MR associated with the node, because the MR makes a registration request process to the HA when it moves to a different location. Therefore, the HA tunnels to an MR's CoA and the MR forwards the data from the HA to the nodes in its network.

2.3 The RO Scheme Based on Path Control Header

The RO scheme implemented in the Path Control Header (PCH) [2] reduces the routing overhead through a tunnel between an MR managing a VMN's link and a Correspondent Router (CR). The CR is a router that makes use of the optimized routing tunnel, using a 3-way handshake and playing the role as a gateway in the CN's domain. For the first phase, the PCH scheme is based on piggy-backing, which carries address information as shown in Fig. 1, and each MR's HA piggybacks its MR's CoA while the first packet passes each HA through tunnels. It requires a new infrastructure [7], *i.e.*, CR. And it creates a tunnel between the CR and MR. The advantage of the PCH scheme is that, once the first packet

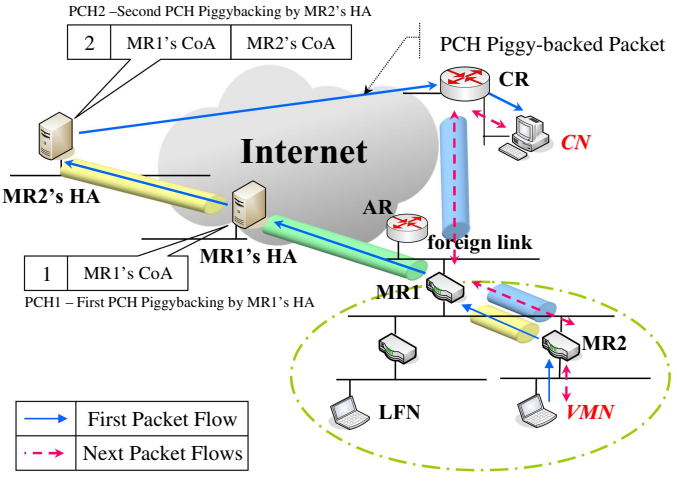


Fig. 1. The Path Control Header (PCH) Scheme

arrives at a CR through pinball routing, the CR could make a RO tunnel and use the tunnel directly for the duration of the session.

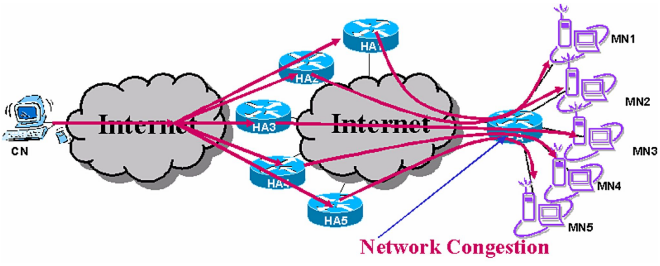
3 Proposed Similar Source-Based Routing Multicast

3.1 Motivation for Multicasting in NEMO

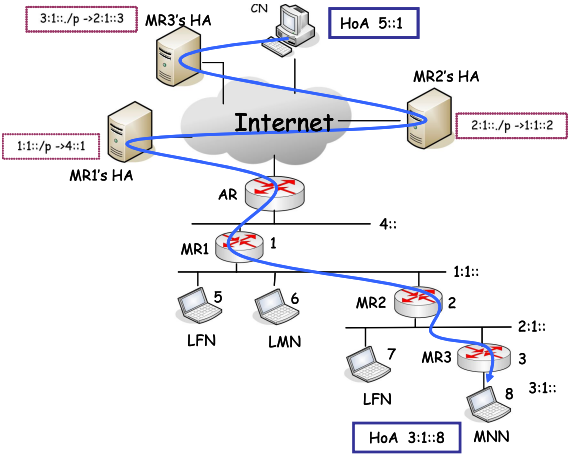
Since NBS is based on MIP, when used to support multicast for NEMO, it inherits many serious problems as MIP [4]. The major problems are the tunnel convergence problem (see Fig. 2(a)), and RO problem (see Fig. 2(b)). In Fig. 2(a), when many MNs, belonging to different HAs, move to the same foreign network, each of the HAs establishes a separate tunnel to the Access Router (AR). If all of the MNs were subscribed to the same multicast group, all tunnels would carry the same multicast packet, resulting in packet duplication, and this is called the tunnel convergence problem.

Furthermore, because the NBS is using bi-directional tunnels, multicasting on NEMO suffers from a weakness, referred to as the pinball routing problem, as in Fig. 2(b), which can also be classified as an RO problem. Fig. 2(b) shows an example of the pinball routing problem when 3 levels of nesting exist. In this case, the original data is encapsulated three times. MR1, MR2, and MR3 decapsulates the packet and forwards the packet to the Mobile Network Node (MNN, *i.e.*, LFN, Local MN (LMN), and Visiting MN (VMN) [8]). If a MONET has N levels of nesting, the data is encapsulated N times. The overhead of pinball routing becomes more significant as the nesting level increases and the distance between the HAs of MRs becomes longer.

If the BT is adopted on NEMO, the tunnel convergence problem is occurred and the problem gives much overhead not only on MRs but also on the AR.



(a) Tunnel convergence problem



(b) Pinball routing problem

Fig. 2. Tunnel convergence and pinball routing problems

Because they must process many tunnels, the problem incurs load only to AR or mobility agent in MIP. In addition, the more networks are nested, the more damage occurs in the entire MONET. If the RS is adopted on NEMO, MNNs in the MONET cannot join a multicast group, because they are not aware of where the multicast routers are located. In NBS architecture, bi-directional tunnel is used to enable an MR to communicate with its home network. However, this causes the serious RO problem, *i.e.*, the pinball routing problem as presented in Fig. 2(b). To overcome this weaknesses in multicasting, we should adopt PCH [2] as RO solution. The pinball routing problem is solved by the RO solution, furthermore, we propose the new scheme for solving the tunnel convergence problem in nested mobile networks. We introduce to the proposed scheme in the next subsection.

3.2 Operation for the Proposed Multicast Scheme Based on PCH

There are some assumptions. It is assumed that in the proposed architecture, multicast functions are built into MRs and AR. And the AR already is joined

to multicast tree. In addition, the RO protocol PCH is used in NEMO environments. As shown in Fig. 3, every multicast group member, MR1 and MR4, has been receiving the multicast services. Multicast member MR6 is attached to the MR5. When MR6 attaches a new link behind the MR5, CR has to know the existence of nested tunnels through PCH information. At this time, Binding Request (BR) message contains Nested Routing Path (NRP) option. NRP option is used to inform MR6 of the nested path information. If AR receives BR message having NRP option, AR also gets know the topology of nested mobile networks and MR6 wanted to multicast service. Therefore, the AR should make an adjacency matrix, Fig. 4(b), for multicasting in the nested mobile network. Actually, the adjacency matrix is a tree for multicast services.

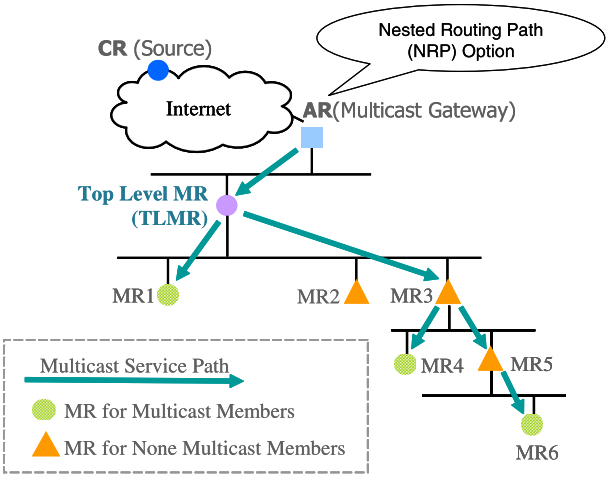
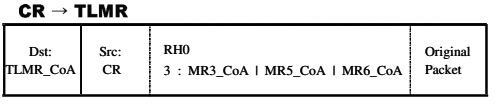
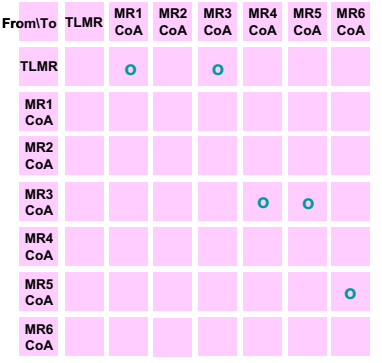


Fig. 3. Example for multicasting based on PCH

In PCH RO mechanism, the routing path information is written in the packet header as in Fig. 4(a). Therefore, the adjacency matrix must be maintained at AR, and the matrix is updated by BR messages. As shown in Fig. 5(a), it is assumed that the MR3 moves to outside the foreign network. Hence, the MR4 and MR6 do not receive multicast packets temporarily. In the currently existing schemes, MR4, MR5, and MR6 - which are all under MR3 - may perform Binding Update (BU) to their HAs and the multicast tree is reconstructed. For this reason, the traffic overhead for continuing multicast service and tree reconstruction time increases. When the MR detects multicast service interruption, the MR becomes aware that something is wrong. At this time, the MR expects to receive the Router Advertisement (RA) message. If it receives the message, then its new CoA address will be registered by new BR message. At the AR, the adjacency matrix may be updated. As described in Fig. 5(b), however, the matrix is not tree because the components are two things in 5th and 6th columns. The

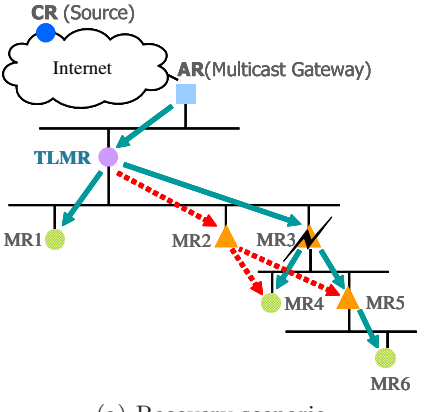


(a) Packet delivery via nested RO tunnel from CR to TLMR in BR message

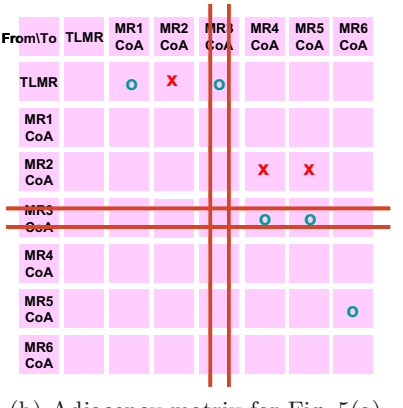


(b) Adjacency matrix for Fig. 3

Fig. 4. Similar source based routing based on PCH at AR



(a) Recovery scenario



(b) Adjacency matrix for Fig. 5(a)

Fig. 5. Updating the adjacency matrix at AR

components in 4th row are removed and it also gets rid of 4th column component. Hence, the AR can obtain the new tree from the updated adjacency matrix.

4 Performance Evaluation

As a matter of convenience, the proposed scheme is named that the PCHm scheme. Now, the multicast performance of the schemes mentioned in Section 2 is calculated. The multicast tree delay variables for providing multicast among BT, RS, and PCHm mechanisms are compared. The parameters required to compute multicast tree delay variables are stated belows:

- D_w : Number of average hops between HAs in the wired network (outside of the AR's network)
- D_n : Number of average hops of a visited network (under the AR's wireless network)
- L_{ps} : Processing time
- L_{pg} : Propagation time
- B_w : Wired transmission speed
- B_n : Wireless transmission speed
- S_{pkt} : Packet size
- τ : The adjacency matrix update-time at AR
- l : Nesting level of an MR counting from the AR

The single hop delay notations are simplified as follows:

$$T_w = L_{ps} + \frac{S_{pkt}}{B_w} + L_{pg} \tag{1}$$

$$T_n = L_{ps} + \frac{S_{pkt}}{B_n} + L_{pg}. \tag{2}$$

Table 1. Parameters value for analysis [9][10][11]

D_w	D_n	L_{ps}	L_{pg}
10	1	0.5 $msec$	2 $msec$
B_w	B_n	S_{pkt}	τ
155 $Mbps$	144 $Kbps$	1480 $Byte$	1 $msec$

Several parameters are set to calculate the numerical results of each scheme. The nested level l is from 1 to 8. The parameters are used in Table 1. The tree reconstruction delay, φ_D, functions are the following:

$$\varphi_D(BT) = T_w D_w(2 + l) + T_n D_n l \tag{3}$$

$$\varphi_D(RS) = T_w D_w(1 + l) + T_n D_n l \tag{4}$$

$$\varphi_D(PCHm) = T_n D_n l + \tau \tag{5}$$

After an error occurs at the nested level l, if the multicast NEMO protocols perform all BUs, then multicast service continuity is unavailable. Hence, the multicast recovery time must be reduced in order to continue multicast service. In Fig. 6, $\varphi_D(PCHm)$ reaches approximately 25% improvement over $\varphi_D(RS)$ at nested level 8. In fact, our proposed PCHm has an extremely fast recovery time, compared to other well-known schemes.

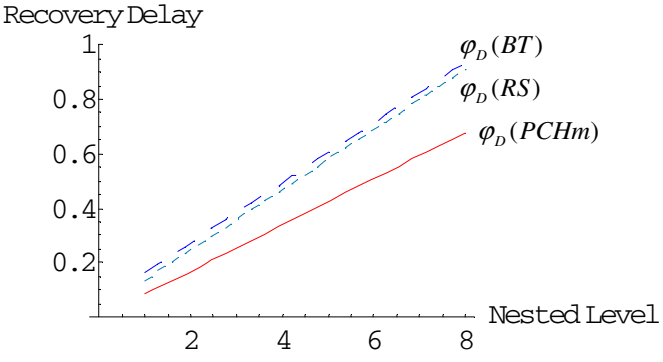


Fig. 6. Recovery times

5 Conclusion

Since NBS architecture is based on MIP, it inherits the same fundamental problems from MIP, such as the tunnel convergence problem, when it is used to support multicast for MONETs. In this paper, the provision of multicast services using the PCH [2] and the proposed scheme providing efficient and optimized route paths for multicasting in NEMO are examined. Furthermore, in real life networks, such failures frequently occur more often than expected. In the case that a node or link constituting a multicast tree fails, multicast members of the subtree rooted at the failed or migrated node cannot receive message. In order to support seamless and real time service, multicast routing should be reliable so that is a simple but very efficient mechanism to rapidly recover from failures. In this paper, we also introduce a very simple method for reducing multicast recovery time in NEMO. Since the proposed scheme provides optimal and efficient multicast services and does not have the pinball routing problem in MONET environments, the proposed architecture can serve a basic multicast support mechanism in NEMO environments.

References

1. Devarapalli, V., Wakikawa, R., Petrescu, A., Thubert, P.: Network Mobility (NEMO) Basic Support Protocol. RFC 3963. IETF (2005)
2. Na, J., Cho, S., Kim, C., Lee, S., Kang, H., Koo, C.: Route Optimization Scheme based on Path Control Header. Internet Draft, draft-na-nemo-path-control-header-00.txt. Internet Engineering Task Force (2004)
3. Perkins, C.: IP Mobility Support. RFC 2002. Internet Engineering Task Force (1996)
4. Johnson, D. B., Perkins, C.: Mobility Support in IPv6. Internet Draft, draft-ietf-mobileip-ipv6-12.txt. Internet Engineering Task Force (2000)

5. Janneteau, C., Petrescu, A., Olivereau, A., Lach, H.-Y.: IPv6 Multicast for Mobile Networks with MLD-Proxy. Internet Draft, draft-janneteau-nemo-multicast-mldproxy-00.txt. Internet Engineering Task Force (2004)
6. Montenegro, G.: Bi-directional tunneling for Mobile IP. Internet Draft, draft-montenegro-tunneling-00.txt. Internet Engineering Task Force (1996)
7. Wakikawa, R., Koshiba, S., Uehara, K., Murai, J.: ORC: Optimized Route Cache Management Protocol for Network Mobility. In: Proceeding of ICT, vol. 2, pp. 1194–1200. IEEE Computer Society Press, Los Alamitos (2003)
8. Ernst, T., Lach, H.-Y.: Network Mobility Support Terminology. Internet Draft, draft-ietf-nemo-terminology-02.txt. Internet Engineering Task Force (2004)
9. Lim, H.-J., Lee, D.-Y., Kim, T.-K., Chung, T.-M.: A Model and Evaluation of Route Optimization in Nested NEMO Environment. IEICE Transaction on Communication, E88-B(7) IEICE, 2765–2776 (2005)
10. McNair, J., Akyildiz, I.F., Bender, M.D.: An inter-system handoff technique for the IMT-2000 system. In: Proceeding of INFOCOM, pp. 208–216. IEEE Computer Society Press, Los Alamitos (2000)
11. Lampson, B., Srinivasan, V., Varghese, G.: IP lookups using multiway and multicolumn search. In: IEEE/ACM Transactions Networking, vol. 17, pp. 324–334. IEEE/ACM, New York (1999)

An Effective XML-Based Sensor Data Stream Processing Middleware for Ubiquitous Service

Hun Soon Lee[1] and Seung Il Jin[2]

[1] Internet Server Group, Electronics and Telecommunications Research Institute
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea
hunsoon@etri.re.kr
[2] Division of Electrical and Computer Engineering, Chungnam National University
220 Gung-dong, Yuseong-gu, Daejeon, 305-764, Korea
sijin@cnu.ac.kr

Abstract. As sensor-related technologies have improved, ubiquitous services which give convenience to people using them are increasing more and more. It is necessary to make service developer's best endeavors to provide these services. In this paper, we propose a novel XML-based sensor data stream processing middleware called UbiCore (Ubiquitous Core) which helps service developer to make ubiquitous service easily. UbiCore manages scattered diverse sensors, collects data from sensors, processes collected sensor data, and automatically delivers the processed results to the related back-end applications. We can save the developing cost of ubiquitous service by using the proposed middleware.

Keywords: ubiquitous computing, middleware, sensor data, continuous query, context-driven service.

1 Introduction

One of the core keyword for the next generation computing environment is "ubiquitous." Recently, researches on ubiquitous computing are proceeding actively. Sensor-related technologies such as radio frequency identification (RFID), sensor and sensor networking are paid attention to the key enabling technologies for building a ubiquitous computing lifestyle. Moreover, these technologies have been increasingly used in ubiquitous applications such as supply chain management, retail stores, healthcare, industrial automation, security, disaster protection, asset management and so on.

In the ubiquitous computing environment, various sensors having computing ability are scattered throughout our surroundings. These sensors generate information – called sensor data – endlessly like water flowing. To extract meaningful information from the sensor data and to be able to use it, complex and rapid processing is needed. A middleware system which acts as a bridge between sensors and application services is essential to enable application developers to easily develop ubiquitous service. This middleware should be able to manage diverse sensors and to collect data from sensors. It also should be able to extract meaningful information from collected data

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 844–857, 2007.
© Springer-Verlag Berlin Heidelberg 2007

and to deliver the information into application systems which have an interest in this information.

Recently, Several middleware systems handling sensor data such as Sun's Java System RFID Software, Oracle's Sensor Edge Server, CapTech's TagsWare, ConnecTerra's RFTagAware, TelegraphCQ[1], STREAM[2], and Aurora[3] have been proposed and some of them are commercially available. But the above mentioned middleware systems insufficiently consider applications using various formats of sensor data and the complex filters. Therefore, it is necessary to make application developer's best endeavors to develop these applications.

In this paper, we propose a novel middleware system for sensor data stream processing called UbiCore (Ubiquitous Core). The major contributions of UbiCore can be summarized as follows. First, it offers its own query language called XQueryStream (XQuery for Stream Data) in order to incorporate processing capabilities of sensor data stream. Second, it accelerates the widow query processing by efficiently managing the intermediate result of previous query evaluation. Third, it supports query on both continuously generated stream data and archived historical data. Forth, it adopts the preprocessing phase called pre-filtering prior to the query evaluation to speed up the processing of sensor data stream. Fifth, it offers a distinct markup language called Context-driven Service Markup Language (CSML) to easily specify the linking information between a context and a service.

This paper is organized as follows. Section 2 overviews our novel XML-based middleware system for sensor data stream processing named UbiCore. Section 3 describes the details of proposed middleware. Section 4 explains experiments. In section 5, we discuss related work. Finally in section 6, we present conclusions and future works.

2 XML-Based Sensor Data Stream Processing Middleware

Recently, XML is being used as the standard of data interchange over the internet. Also, it is expected that data generated by sensor will be transmitted to middleware in the form of XML. In this paper, we designed an XML-based sensor data stream processing middleware called UbiCore.

UbiCore has following features:

Gearing with diverse sensor. UbiCore can operate with not only UHF-band RFID readers but also sensors which sense temperature, humidity, and so on. Moreover, it can operate with the sink node of a wireless sensor network. UbiCore has extensible driver architecture to easily develop and implement custom drivers to plug in to it.

Processing the diverse formats of sensor data streams. UbiCore allows users to express the structure of sensor data using XML schema[4] and to query based on this structure.

Supporting near real-time processing of sensor data streams. UbiCore endeavors to support near real-time processing of sensor data streams by having the preprocessing phase called pre-filtering and reusing the intermediate result.

Supporting context-driven service. UbiCore supports context-driven services which will be automatically called when a specified context is satisfied. Continuous query language XQueryStream is designed to express context. UbiCore supports integrated query on both continuously generated stream data and archived static data. The service which will be called when the specified context is satisfied may be a web service, a java message service, a HTTP service, a file service, a TCP service, or a logging service. It is easy to add a new service type. The linking information between a specific condition and a back-end service is specified using CSML.

Managing meaningful sensed data. UbiCore supports storing the meaningful XML[5] data stream on a relational database management system (RDBMS). It also enables users to retrieve stored data employing XQJ (XQuery API for Java)[6] and XQuery.

Access control on sensor data stream. UbiCore has an access control mechanism based on the structure and value of the XML data.

Providing Java APIs. UbiCore supports the native Java interfaces and ALE[7] interfaces for application developers.

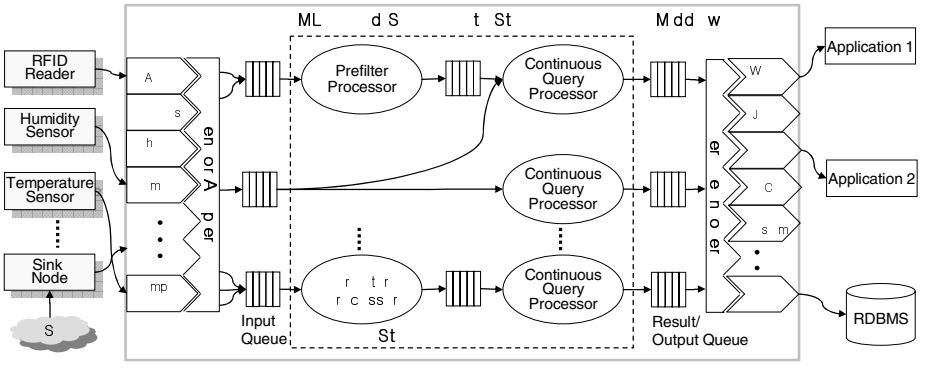


Fig. 1. Processing flow of sensor data in UbiCore

The processing sequence for each sensor data is shown in Fig. 1. Sensed data is transformed to a message and this message is transmitted to the middleware (UbiCore) which is connected with the corresponding sensor. In UbiCore, the sensor adapter which is waiting for the arrival of sensed data receives this message. The sensor adapter puts it into the input queue which is related to this sensor. This input queue was created when we logically grouped several sensors into a data source. If a pre-filter is defined, the noisy sensor data stream is refined by removing the sensor data which satisfies the pre-filter condition. After that, the continuous query processor extracts meaningful data from the sensed and pre-filtered data stream which satisfies user interest. Then, it transforms the extracted data into the customized format which is specified by the user in an XQueryStream statement. After that, the transformed result is put into the output queue. The service invoker de-queues the data

from the output queue and invokes an application using this data as a parameter or stores it into the RDBMS in order to use it later.

3 Details of Proposed Middleware

In this section, we describe the details of our proposed system for ubiquitous service.

3.1 Continuous Query Language: XQueryStream

Middleware for sensor data stream processing should be able to capture meaningful information from sensor data streams. Therefore, a query language to describe the characteristics of interesting data is needed. Queries over a point-in-time snapshot of the data set like a traditional DBMS query are evaluated once, but queries on data streams are evaluated continuously as data streams continue to arrive[8]. This kind of query is called continuous query. In a continuous query, a window concept is used to restrict the amount of data that participates in the query over time. Generally, time or the number of events (tuples) is used as a measure to restrict the input data.
In UbiCore, continuous query on XML-formed sensor data is expressed using its own query language, called XQueryStream which originates from XQuery[9]. XQueryStream is extended to allow user to define the window to restrict the data that participates in the query from streamed sensor data. XQueryStream can express the complex filters about sensor data stream whose format is defined by using the XML schema because it is based on XQuery.

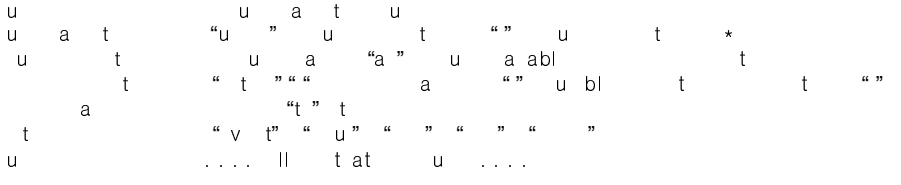


Fig. 2. A part of grammar for XQueryStream

A part of grammar about XQueryStream is expressed in Extended Backus-Naur Form (EBNF) [10] notation as in Fig. 2. An XQueryStream statement is defined by two terms: a query target <QueryTarget> and a query condition <QueryBody>. The user limits the data which will be used as an input of query evaluation using a query target <QueryTarget> and specifies the user's interest using a query condition <QueryBody>. The syntax to describe the query condition <QueryBody> follows that of XQuery [9]. <QueryTarget> has source definition parts <SourceDefinition> more than one following the keyword “using.” <SourceDefinition> is defined by three terms: a source name <SourceName>, a source variable name <SourceVariable> and a window definition <WindowDefinition>. <SourceName> is a name of the data source which was used when we logically grouped sensors. <WindowDefinition> consists of the window range <windowRange> which determines a window region over a given stream, the length of tumbling <TumblingLength> and a unit of window

<Unit>. A window is repeatedly established for each instance of a given duration which is specified by <TumblingLength>. Both <WindowRange> and <TumblingLength> should be given in terms of time or event which is chosen by <Unit>. If a query does not have window definition, it will be evaluated in every event.

A simple example query with a sliding window is following.

Example 1. Every five minutes, retrieve the number of selling items over the last 10 minutes.

```
using "SellingItemStream" as $src
          within (0 to 10, 5, min)
<NumItem> { count($src/Tag) } </NumItem>
```

3.2 Efficient Evaluation of Window Query

Data can be included several times as a result of window binding if the window used for a query is a sliding window with overlap. The result of window binding is used as an input of query condition evaluation. So if we do not specially care it, we may evaluate the same query multiple times on equal data. Fig.3 depicts an example of window binding for a query which has a sliding window with overlap. Data which comes within interval (t+4, t+5) is included 5 times as a result of window binding and used for query evaluation. It is an intolerable waste of computing power that the same operation is performed multiple times on the same data.

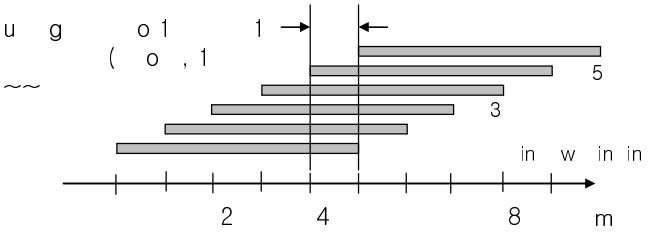


Fig. 3. An example of window binding for a query which has a sliding window with overlap

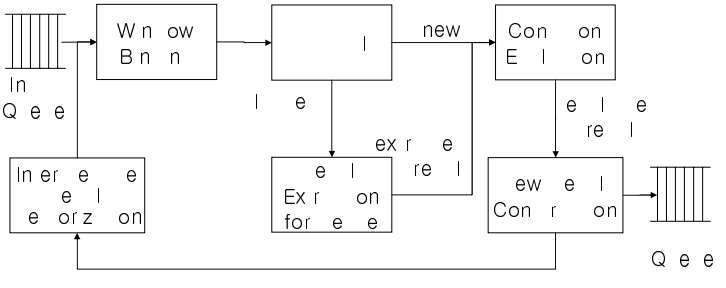


Fig. 4. Procedure of window query evaluation

The XQueryStream processing engine in UbiCore minimizes the unnecessary query evaluation on already evaluated data by efficiently managing the result of previous query execution. The XQueryStream processing engine memorizes the intermediate result of query execution and uses it if the same query evaluation is performed on the same data. It has the advantage of improvement of overall system performance.

Fig. 4 depicts the procedure of window query evaluation which reuses intermediate results. Window binding is performed on the sensed data which are in the input queue. After that, we split this data into two parts using window definition: one is the data which is contained in the result of previous window binding and the other is the data which is newly included in the result of this window binding. We can use the intermediate result of a previous query evaluation if this query has a sliding window with overlap. To prevent a query from being evaluated multiple times on same data, we extract the intermediate result of the query condition evaluation from the memorized result of the previous evaluation of the query condition if there is duplicated data (which is contained in the result of both the previous window binding and this window binding) as a result of the data split. The evaluation of the query condition is performed using newly arrived data and extracted result as an input. The XQueryStream processing engine makes query result with them and passes it to the output queue. The intermediate result of query condition - such as results of path expression evaluation, results of aggregation function evaluation, and so on - is memorized for the purpose of use it later. It is a continuous query, so the above procedures are repeated until the query is withdrawn.

3.3 Hybrid Query

Recently, many applications require a real-time processing of queries on both continuously generated stream data and historical data which is stored in the persistent DBMS. We define this kind of queries as a hybrid query. In order to handle such hybrid queries, a system must be capable of supporting an integrated query processing for persistent data residing on databases and for transient data flowing through the wires or the wirelesses. The particularly significant problem to support hybrid queries is the I/O cost of fetching archived historical data. Given the dramatic improvements in computation power and network bandwidth, the cost of interleaving random historical data accesses with the processing of live network data has become substantial. These I/O costs can have a debilitating effect on the ability of a query processing engine to run multiple hybrid queries.

To minimize the I/O costs, it is necessary to reduce the number of evaluation and historical data size for a query without modifying the archived historical data. In order to resolve this problem, the query processing engine in UbiCore executes hybrid queries on cached data which are results generated by XML triggers. The XML trigger is to specify select operations for retrieving interesting data as the trigger action whenever associated historical data is changed. This XML trigger mechanism is essential for efficient hybrid query processing, because XML triggers independently have performed portions of hybrid queries in advance. Fig. 5 depicts the key concepts for hybrid query processing. Our XML triggers need to be translated into SQL triggers in relational databases. The SQL trigger results stored in relational databases

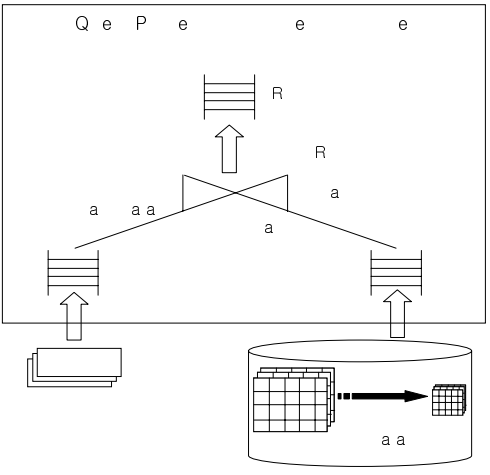


Fig. 5. Key concepts for hybrid query processing

should be translated into XML documents, and should be cached in the execution space of the continuous query. Then, the query processing engine should use the results to process the integrated query.

3.4 Pre-filtering

In the ubiquitous computing environment, data sensed with one sensor can be used as an input of one or more continuous query for ubiquitous services. The amount of work to process the continuous query will be reduced if they are fed to the continuous query processor after some sensed data which can not be results of continuous queries are filtered out in advance. For this reason, UbiCore has a pre-processing phase prior to query evaluation. It prevents unnecessary data from being used as an input of continuous queries. This pre-processing phase uses a software filter. We call it a pre-filter which expresses the characteristics of data to be removed. We can express pre-filter using XQueryStream. The pre-filter adopts the syntax of XQueryStream but gives a different semantics. Data which satisfies query conditions will be returned if a query statement is used as a continuous query, but filtered out if a query statement is used as a pre-filter. A query for pre-filtering should produce valid document on original schema of data source. The XQueryStream processing engine with pre-filtering phase is illustrated in Fig. 1. The XQueryStream processing engine consists of a pre-filter processor and a continuous query processor. In advance, the sensor data is refined by the pre-filter processor. After that, the continuous query processor extracts data relevant to the user's interest from the pre-filtered data, makes it into the form of user wants and passes it to the query result queue. One data source can feed data to multiple continuous queries, but can have only one pre-filter. Pre-filtering seems to have little effect to enhance the performance from the viewpoint of single continuous query processing; however, we can get great performance enhancement because the input data of the continuous query will be decreased and this data will be used as an input for multiple – dozens or hundreds of – continuous queries.

3.5 Specifying the Context-Driven Service

UbiCore supports context-driven services which will be automatically called when the given context is satisfied. The application developer has to notify the information relating to the being provided service to a middleware so that the context-driven service can be provided by using UbiCore. To make application developer easily express this information, UbiCore offers a markup language called CSML. Information for describing the context-driven service comprises 5 components as depicted in Fig.6. <ContextInfo> contains information not only to recognize the context when the service will be invoked but also to get the dynamic input values which will be fed to service as a parameter. It is represented using XQueryStream. <ServiceInfo> holds information about service which will be called when the specified <ContextInfo> is satisfied. Services we can currently use with UbiCore are Web Services, JMS, HTTP services, file services, TCP services and logging services. And we can easily add a new service type. <MappingRuleInfo> expresses information for connecting a given context and a given service such as the mapping rules between parameter and extracted data, the period of service invocation, and so on. <LifeTimeInfo> represents information about the time that context-driven service has

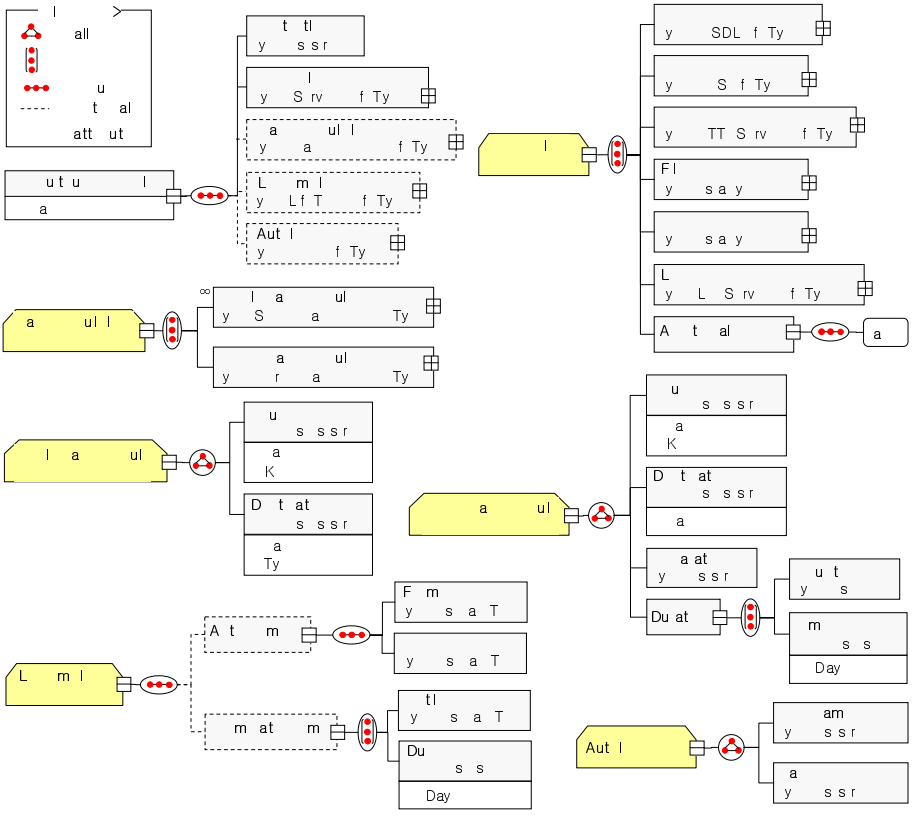


Fig. 6. A part of information structure for specifying the context-driven service

```
<UbiquitousServiceInfo>
 <ContextInfo><![CDATA[
  using "DS_Entrance" as $src
  for $cart in $src/Cart
  return
   <Result>
    <CartID> {$cart/CartID/text()} </CartID>
   </Result>
  ]]>
 </ContextInfo>
 <ServiceInfo>
  <WebService>
   <WSDLInfo>
    <WSDL-URL> http://192.168.1.10:8080/UShop.wsdl </WSDL-URL>
    <ServiceName>Coupon</ServiceName>
    <PortName>UShop</PortName>
    <OperationName> issueDiscountCoupon </OperationName>
   </WSDLInfo>
  </WebService>
 </ServiceInfo>
 <MappingRuleInfo>
  <SingleMappingRule>
   <Source value="/Result/CartID" kind="dynamic" />
   <Destination name="CartId" type="string"/>
  </SingleMappingRule>
  <SingleMappingRule>
   <Source value="10" kind="static" />
   <Destination name="DiscountRate" type="long"/>
  </SingleMappingRule>
 </MappingRuleInfo>
 <LifeTimeInfo>
  <ActiveTime>
   <From>10:00:00</From> <To>11:00:00 </To>
  </ActiveTime>
  <TerminationTime>
   <Until> 2007-04-30T24:00:00.000+09:00 </Until>
  </TerminationTime>
 </LifeTimeInfo>
</UbiquitousServiceInfo>
```

Fig. 7. An example of CSML for issuing money-off coupon in ubiquitous shop

meaning. And <AutoInfo> holds information which is needed to check the authorized use of given service. An example of CSML for a web service which issues money-off coupons when a customer visits a shop is depicted in Fig. 7.

4 Experiments

UbiCore was developed to meet the goal which is to support the 3,000 service calls per second in the environment in which 30,000 sensed data per second is inputted and 200 context-driven services are registered. We implemented the prototype of UbiCore using Java in the J2SE 5.0 JDK environment. And then, to measure the UbiCore satisfying the above mentioned goal, we construct experimental environment as shown in Fig. 8. It has five components such as sensor emulators, a load generator, a UbiCore, a main memory-resident database management system(MMDBMS), and a result analyzer. The sensor emulators emulate real RFID sensors. They generate RFID tag data which is EPC data encoded using SGTIN-64. The load generator plays the role of sensor registration, data source creation, context-service registration, and so on. Continuous queries which filter out the 80%, 85%, 90%, and 95% - respectively a quarter, i.e., 50 – of sensed data were used to express the context. To easily

know the number of service calls, we use the logging service which inserts a record into the MMDBMS whenever an EPC tag data which satisfying the context is detected. Log record consists of EPC value, the time in which data came into the UbiCore(EnteringTime), and the time in which logging service is called(LoggingTime). The result analyzer issues queries to the MMDBMS in order to gather the experimental result once every 5 seconds. The response time can be calculated using the logged data, that is, the gap between EnteringTime and LoggingTime. And the number of service call can be calculated by counting the number of logged records.

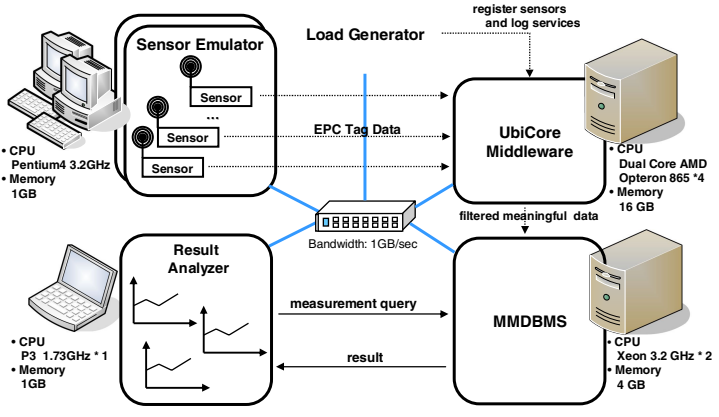


Fig. 8. Experimental environment

Table 1 shows detailed our experimental environment and result. We used the continuous queries which extract 20%, 15%, 10%, and 5% - respectively a quarter – of inputted data to express the context. So, the expected total number of service calls per second is 3750(30,000 X (0.2+0.15+0.1+0.05)/4). Our experimental result shows that the total number of service calls per second which was called within 1 second after tag data was inputted is 3732 which is 99.53% of our expectation. We sampled 100 times - once every 5 seconds - to get the experimental result. Therefore, we can conclude that UbiCore processes sensed data in near real-time(average response time is 16 msec) and invokes the services as many as needed in our objective environment.

Table 1. Experiment at the objective environment

	Factors	Value
Load	number of sensor emulators	400 EA
	interval of data generation	1 sec
	number of sensors per data source	2 EA
	number of tags per document(message)	75 EA
	number of tags per second	30,000 EA
	number of context-driven services	200 EA
Result	number of service calls per second (response time < 1 sec)	3732 (99.53 %)
	average response time	16 msec

We also measured how the performance of UbiCore changed against the number of tags in a message. For an experiment, we fixed the number of data sources, the number of context-driven service, the number of sensors, and the number of inputted tags per second. After that, we measured the performance while diversifying the number of tags in a message to 50, 100, 200, and 400. Table 2 shows detailed our experimental environment and the results. It was found from the result that the average response time was also increased as the number of tags within a message grows. The reason why the average response time was increased as the size of a message enlarged was that UbiCore parsed the inputted data using DOM to process the query. The garbage collection of JVM was frequently carried out because DOM parsing of inputted data used much memory. And query processing thread was blocked while the garbage collection was performed; therefore, the average response time was increased.

Table 2. Performance experiments against the number of tags in a message

	Factors	Value			
		Test1	Test2	Test3	Test4
Load	number of sensor emulators	300 EA			
	interval of data generation	0.5 sec	1 sec	2 sec	4 sec
	number of sensors per data source	3 EA			
	number of tags per document(message)	50 EA	100 EA	200 EA	400EA
	number of tags per second	30,000 EA			
	number of context-driven services	100 EA			
Result	number of service calls per second (response time < 1 sec)	3740 (99.74)	3745 (99.86%)	3670 (97.85%)	2935 (78.28%)
	average response time	8 msec	24 msec	63 msec	98 msec

Table 3. Performance experiments against the number of sensors

	Factors	Value		
		Test5	Test6	Test7
Load	number of sensor emulators	100 EA	300 EA	600 EA
	interval of data generation	1 sec		
	number of sensors per data source	1 EA	3 EA	6 EA
	number of tags per document(message)	300 EA	100 EA	50 EA
	number of tags per second	30,000 EA		
	number of context-driven services	100 EA		
Result	number of service calls per second (response time < 1 sec)	3490 (93.06%)	3745 (99.86%)	3750 (100%)
	average response time	105 msec	24 msec	29 msec

In addition, we experimented how the performance of UbiCore changed against the number of sensors. For this experiment, we increased the number of sensors to 100,

300, and 600 while fixing the other factors such as the number of data sources, the number of context-driven service, and the number of inputted tags per second. In this experimental environment, increasing the number of sensors means that the number of tags in a message is decreased. Detailed our experimental environment and the results are shown Table 3. It seems that the increment of the number of sensors nearly did not have an effect on the performance. But, it is because the deterioration of performance due to the increment of the number of sensors was offset by the improvement of performance due to the reduction of the number of tag data coming in from one sensor.

5 Related Work

Middleware system for ubiquitous application using sensor-related technologies should be able to handle various informal data streams, to transform the meaningful data to the form which is used in an application, and to relay the transformed meaningful data to the application using various service protocols.

There are several middleware systems and solutions for sensor data stream processing that have been proposed and some of them are commercially available, but none of them satisfy the above mentioned requirements.

Table 4. Comparison among middleware from qualitative perspectives

	UbiCore	STREAM	Java System RFID Software	Sensor Edge Server
Motivating Applications	all-purpose	all-purpose	RFID	all-purpose
Input Format	XML-formed	relational	relational	relational
Allowed input	stream and historical data	stream and historical data	stream only	stream only
Query Language	XQueryStream	CQL	none	none
ALE support	yes	no	yes	yes
Expressiveness of context	rich	rich	poor	poor
Data transformation	rich	poor	middle	middle
Specifying Service lifetime	Support using CSML	Not support	Not support	Not support
Service	File, JMS, HTTP, Logging, TCP, Web Services	Logging, Stream	File, JMS, HTTP, TCP, Web Services	File, JMS, HTTP, Logging, TCP, Web Services

Several middleware systems - such as Sun's Java System RFID Software, Oracle's Sensor Edge Server, CapTech's TagsWare, ConnecTerra's RFTagAware - which are specialized to handle RFID tag data stream are available commercially. They can

deliver meaningful sensor data into applications using various service protocols such as JMS, HTTP, Web Services, and so on. They support ALE interface, but they have no query language. Users can express what they want using the connection of simple filters; therefore, the expressiveness of the filters is poor. Besides, they have lost sight of the important fact that sensor data streams can have diverse complex formats because they are specialized to handle RFID tag data stream. They do not support queries on archived historical data.

Early version of UbiCore [11] which is specialized for RFID application has been developed. It is insufficiently considered about diverse sensor data which are generated various kinds of sensors. It only operates with RFID readers. And only four kinds of services - i.e., a web service, a java message service, a HTTP service, and a log service - can be used as a context-driven service. Besides, it is hard to add a new sensor type and a new service type which will be supported.

In addition, data stream processing systems - such as TelegraphCQ [1], STREAM [2], and Aurora [3] - have been developed by the academic database research community. They extract the meaningful data from raw data stream and generate the stream type results or store them in the repository such as DBMS. However, the manner for delivering a result with various services is not considered. TelegraphCQ and STREAM can execute queries over historical data in addition to data streams with a relational model by using SQL-like query language. Aurora enables users to express what they want using network of relational operator.

Table 4 shows a comparison among middleware from qualitative perspectives.

6 Conclusions and Future Works

In this paper, we have presented an effective XML-based middleware system for sensor data stream processing called UbiCore. UbiCore has its own query language called XQueryStream which originates from XQuery in order to incorporate processing capabilities of sensor data streams. And, UbiCore has been enlarged to query on both live stream data and archived historical data based on the XML trigger. To speed up the sensor data stream processing, UbiCore performs a preprocessing called pre-filtering prior to query processing and reuses the intermediate result of previous query evaluation to process the query. In addition, UbiCore has a distinct markup language called Context-driven Service Markup Language which can easily specify the linking information between a context and a service.

We are now making our best endeavors to increase the performance of UbiCore. To find and eliminate the bottleneck point of performance, we are profiling our system. To support multi-query processing, we try to share intermediate results not only within a single query but also among multiple queries.

Acknowledgments. We thank to all members who join the development of UbiCore. This work was supported by the MIC of Korea in the context of the development of the next generation internet server technology project.

References

1. Chandrasekaran, S., Deshpande, A., Franklin, M., Hellerstein, J., Hong, W., Krishnamurthy, S., Madden, S., Raman, V., Reiss, F., Shah, M.: TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. In: Proceeding of CIDR 2003, (January 2003)
2. Motwani, R., Widom, J., Arasu, A., Babcock, B., Babu, S., Datar, M., Manku, G., Olston, C., Rosenstein, J., Varma, R.: Query Processing, Resource Management, and Approximation in a Data Stream Management System. In: Proceeding of CIDR 2003(January 2003)
3. Abadi, D., Carney, D., Cetintemel, U., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N., Zdonik, S.: Aurora: A New Model and Architecture for Data Stream Management. VLDB Journal 2(12) (2003)
4. W3C, XML Schema Part 0, 1, 2, W3C Recommendation (October 28, 2004)
5. W3C, Extensible Markup Language (XML) 1.0 (3rd edn.), W3C Recommendation (February 04, 2004)
6. Michels, J.-E.: XQuery API for Java(XQJ) 1.0, JSR 225 EG Draft Specification (March 03, 2006)
7. EPCglobal, The Application Level Events (ALE) Specification Version 1.0, EPCglobal Ratified Specification (September 15, 2005)
8. Terry, D., Goldberg, D., Nichols, D., Oki, B.: Continuous queries over append-only databases. In: Proceeding of the ACM SIGMOD 1992 (June 1992)
9. W3C, XQuery 1.0 : An XML Query Language, W3C Candidate Recommendation (November 3, 2005)
10. ISO/IEC 4977, Information technology - Syntatic metalanguage - Extended BNF (December 1996)
11. Lee, H.S., Choi, H.H., Kim, B.S., Lee, M.C., Park, J.H., Lee, M.Y., Kim, M.J., Jin, S.I.: UbiCore: An Effective XML-based RFID Middleware System. Journal of KISS: Database 33(6) (2006)

Opportunistic Transmission for Wireless Sensor Networks Under Delay Constraints

Ca Van Phan, Kikyung Baek, and Jeong Geun Kim

Department of Electrical Engineering
Kyung Hee University
Yongin, Republic of Korea
phanvanca@yahoo.com,
apple7925@naver.com,
jg_kim@khu.ac.kr

Abstract. We propose an opportunistic transmission strategy for wireless sensor networks (WSNs) that operate in a strict energy-constrained environment. The WSNs of interest in this work are assumed to host delay-sensitive applications. Our proposed transmission strategy named binary-decision based transmission (BDT) attempts to transmit at good channel conditions while meeting the delay constraint, under time-varying wireless channel. The Markov decision process (MDP) is used to find optimum threshold for transmission decision. Our BDT scheme initiates transmission only when the channel quality exceeds the optimum threshold, so that unsuccessful transmissions causing a waste of energy are avoided whenever possible. Extensive simulations are performed to verify the performance of our proposal over wireless fading channels. The simulation results show that the BDT scheme has 35% higher energy efficiency than its counterparts, thereby further prolonging the network lifetime.

Keywords: Wireless sensor networks, energy efficiency, Markov decision process (MDP), opportunistic transmission.

1 Introduction

Wireless sensor networks (WSNs) are low-cost communication networks that contain a large number of sensor nodes with limited energy [1]. Since the WSN may function until a significant fraction of sensor nodes are operational, energy efficiency is a key technical issue in the design of WSN [2,3,4]. Especially, careful management of energy resources is required to maximize the lifetime of the WSN [5]. The sensor nodes may operate over the time-varying wireless channel whose quality significantly fluctuates over time due to fading and interference. Such time-varying nature of wireless channel imposes many constraints in designing an energy-efficient transmission scheme. For instance, a transmission attempt, when the wireless channel is temporarily bad, is highly likely to be failed and may lead to a waste of energy. To avoid this, the sender may wait until the

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 858–871, 2007.
© Springer-Verlag Berlin Heidelberg 2007

channel becomes better. However, deferring the transmissions until the channel becomes better may cause a longer latency. This is a trade-off problem between energy efficiency and latency. Thus a suitable transmission scheme for the WSNs must be able to adapt to variation of the wireless channel while maintaining a good balance between these two conflicting goals.

In order to improve energy efficiency for WSNs, many different techniques have been recently proposed. Ci et al. in [6] proposed an adaptive adjustment of frame size that is predicted via the extended Kalman filter. In [7], a transmission scheme adopting multicast Ready-to-Send (RTS) and priority-based Clear-to-Send (CTS) was proposed to prioritize the terminal with a good channel in terms of channel access. In [8], the authors obtained an optimal solution of the problem of buffer and channel adaptive transmission for maximizing system throughput. The Receiver Based Auto Rate (RBAR) protocol in [9] allows the receiver to choose the data rate based on the signal-to-noise ratio (SNR) of the RTS packet.

In this paper, we consider an opportunistic transmission strategy for wireless sensor networks hosting delay-sensitive applications. This paper is an extension of our earlier work [10] that took into account the WSNs without delay constraint. Our transmission scheme named binary-decision based transmission (BDT) basically takes an opportunistic transmission approach in which transmissions are attempted only under good channel conditions whenever possible. In the BDT scheme, whether to initiate transmission or not is determined according to the current channel conditions. By exchanging the control messages, e.g., RTS and CTS in 802.11 standard, for channel measurement and associated feedback, the receiver can measure the channel quality and the sender can retrieve such information piggybacked in the return message. To continuously monitor the channel condition, data message and its acknowledgement message are also used to measure and piggyback the channel information. In the BDT scheme, the MDP formulation [12] is used to obtain the optimum threshold for successful transmission. Extensive simulations are performed with ns-2 to verify the performance of our proposal over wireless fading channels. Our transmission algorithms are applied to IEEE 802.11 Distributed Coordination Function (DCF) standard with some necessary modifications [11]. The simulation results show that the use of binary-decision based transmission significantly improves energy efficiency with respect to the plain IEEE 802.11 while satisfying delay requirements.

The remainder of the paper is organized as follows. Section 2 describes our system model. The proposed transmission algorithm and its MDP formulation are given in Section 3. Section 4 describes numerical results and simulations. Finally, we conclude the paper in Section 5.

2 System Model

In this section we give a description on our system model that is concerned with MAC protocol and wireless channel model.

2.1 MAC Protocol

Our transmission schemes are implemented on IEEE 802.11 DCF standard with some necessary modifications. The modifications are associated with channel measurement and a mechanism for its feedback. In this model, RTS/CTS and DATA/ACK messages are all used in pair for continuous channel monitoring and feedback of the measurement results. Although 802.11 MAC protocols may not be suitable for WSNs, we employ it in this work to demonstrate effectiveness of our transmission schemes. Since our schemes require only exchange of short control messages or piggybacking channel-related information in normal data messages, they are readily applicable to different MAC protocols other than IEEE 802.11 DCF protocol.

2.2 Channel Model

A finite-state Markov channel (FSMC) model is used to capture the time-varying behavior of wireless fading channel [13]. In Rayleigh fading channels, the received instantaneous SNR ($\mathbf{y}$) is exponentially distributed with probability density function:

$$f_y(y) = \frac{1}{\rho} e^{-\frac{y}{\rho}}$$

where $\rho = E[\mathbf{y}]$. Let y_i denote the threshold of the received SNR where $0 = y_0 < y_1 < y_2 < \cdots < y_K = \infty$. The channel is said to be in state g_k, $0 \leq k < K$, if the received SNR is in the interval $[y_k, y_{k+1})$. We assume that the transitions in the FSMC model occur at the boundary of time slot in which one fixed-size frame is transmitted and transitions occurs only between adjacent states as shown in Figure 1. Furthermore, the channel gain is constant during one time slot of transmission. The parameters of the Markovian channel can be obtained by using the techniques in [13].

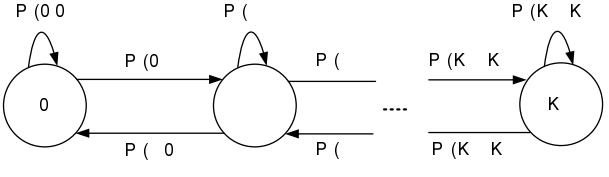


Fig. 1. Finite-state Markov channel model.

We denote $N(y)$ as the level crossing rate given by

$$N(y) = \sqrt{\frac{2\pi y}{\rho}} f_m e^{-y/\rho}$$

where f_m is the maximum Doppler frequency. The state transition probabilities are given by

$$P_g(k,k+1) = N(y_{k+1})\frac{T_f}{\pi_k}, \quad 0 \le k \le K-2$$
$$P_g(k,k-1) \ = N(y_k)\frac{T_f}{\pi_k}, \quad 1 \le k \le K-1$$

where T_f is the frame transmission time and π_k is the steady-state probabilities given by

$$\pi_k = \int_{y_k}^{y_{k+1}} f_y(y)dy$$

For BPSK case, the probability of symbol error $P_b(g_k)$ for state g_k is given by

$$P_b(g_k) = \frac{\delta_k - \delta_{k+1}}{\pi_k} \tag{1}$$

where

$$\delta_k = e^{-\frac{y_k}{\rho}}\left(1 - F\left(\sqrt{2y_k}\right)\right) + \sqrt{\frac{\rho}{\rho+1}}F\left(\sqrt{\frac{2y_k\left(\rho+1\right)}{\rho}}\right)$$

and $F(x)$ denotes the cumulative distribution function (CDF) of a standard normal random variable

$$F(\alpha) = \int_{-\infty}^{\alpha} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$$

Throughout this paper the error statistics follow the BPSK case as formulated above.

2.3 Traffic Model

We assume that the traffic model is a discrete time on-off source with parameters α and β as shown in Fig. 2. The parameters α and β correspond to OFF-to-OFF and ON-to-ON transition probability, respectively. Over time the traffic model alternates between ON and OFF state whose periods follow independent geometric distributions, respectively. Arrival of frame (or packet) occurs each time slot during the ON period only. Delay sensitivity of traffic is modeled by imposing the lifetime of D time slots on each frame. This implies that a frame staying in the buffer longer than D time slots must be dropped. In our analysis, we investigate the impacts of such delay constraint on system throughput. To consider the applications that require up-to-date sensing data under light traffic load, we assume that the buffer can hold at most one frame. Thus any new coming frame preempts the existing one and the existing frame is dropped.

3 Binary-Decision Based Transmission

In this section we describe our transmission algorithm designed to maximize energy efficiency and the MDP formulation.

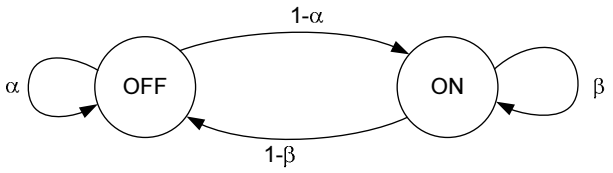


Fig. 2. On-off traffic model with transition probabilities α and β

3.1 Transmission Scheme

In our proposed scheme the sensor node takes either of two actions: *Transmit* and *Defer*, corresponding to transmitting the data and deferring the transmission, respectively. The current wireless channel is measured at the receiver via RTS or Data frame, and is classified into one of two states including *Good* and *Bad* based on the received SNR. This information is notified back to the sender by embedding it in the return frame, i.e., CTS or ACK frame. The SNR threshold used to classify the channel states is determined using the Markov decision process (MDP) as explained later in detail. Figure 3 depicts the BDT scheme with RTS/CTS exchange.

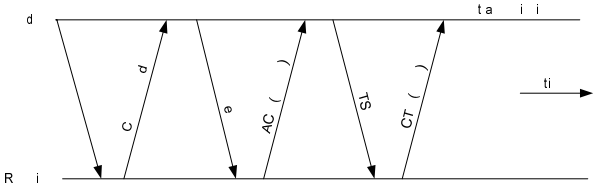


Fig. 3. Binary-decision based transmission

3.2 Markov Decision Process (MDP)

The problem of finding the optimum thresholds for successful transmission is formulated using the MDP. The optimum operation policy in terms of energy efficiency is obtained. For the sake of tractable analysis, it is assumed that time is divided into time slots of equals length of T_f seconds. It is also assumed that the outcome of transmission is immediately available at the end of transmission.

System States. We denote the set of possible system states by S in which S is a finite set. The system state s_i at time slot i is given by

$$s_i = \langle g_i, t_i \rangle$$

where g_i is the channel state at time slot i, $0 \leq g_i < K$, and t_i is the state of sensor node at time slot i. The possible states of the sensor node include *Idle*

and $(D+1)$ *Busy* states where D corresponds to the lifetime of frame. Let I, and B_k $(k = 0, 1, \cdots, D)$ denote *Idle*, and *Busy* states, respectively. The sensor node is said to be in kth *Busy* state (denoted by B_k) when a frame is delayed by k time slots and in *Idle* state when no frame is present.

Control Actions. Depending on system state s_i, sensor node determines to transmit or to defer transmission. Let $A(s_i)$ denote the set of all possible control actions in state s_i. a_i is the control action executed at time slot i. Each action in $A(s_i)$ corresponds to the following values:

$$a_i = \begin{cases} 0, & \text{Defer} \\ 1, & \text{Transmit} \end{cases}$$

Cost of Actions. Given the system state s_i and a control action a_i, the immediate cost incurred in time slot i is defined as

$$C_i(s_i, a_i) = L_e(g_i, a_i)P_t + \delta L_d(s_i, a_i) \tag{2}$$

where P_t is the transmission power set to 1, δ is a weighting factor that indicates a relative importance of frame loss over energy consumption $(0 \leq \delta \leq 1)$, $L_e(g_i, a_i)$ is the expected frame error rate, and $L_d(s_i, a_i)$ is the expected number of frame losses due to buffer overflow or lifetime expiration of the frame. $L_e(g_i, a_i)$ is given by

$$L_e(g_i, a_i) = a_i P_f(g_i) \tag{3}$$

where $P_f(g_i)$ is the frame error rate when the channel state is g_i. Assuming independent bit errors, the frame error rate $P_f(g_i)$ for frame size L and the channel state g_i is given by

$$P_f(g_i) = 1 - (1 - P_b(g_i))^L \tag{4}$$

where $P_b(g_i)$ is obtained from (1). $L_d(s_i, a_i)$ is given by

$$L_d\left(\langle g_i, B_k\rangle, a_i\right) = \begin{cases} \beta\left((1-a_i) + a_i P_f\left(g_i\right)\right), & k = 0 \\ (1-\alpha)\left((1-a_i) + a_i P_f\left(g_i\right)\right), & 0 < k < D \\ (1-a_i) + a_i P_f\left(g_i\right), & k = D \end{cases} \tag{5}$$

System Dynamics. Given the system state $s_i = \langle g_i, t_i\rangle$ and a control action a_i, the probability of the system being state $s_{i+1} = \langle g_{i+1}, t_{i+1}\rangle$ in next time slot is:

$$\Pr\left[s_{i+1} = \langle g_{i+1}, t_{i+1}\rangle \mid s_i = \langle g_i, t_i\rangle, a_i = a\right] = P_g(g_i, g_{i+1})P_t(t_i, t_{i+1}, a)$$

where $P_g(g_i, g_{i+1})$ is the transition probability from channel state g_i to g_{i+1} and $P_t(t_i, t_{i+1}, a)$ is the transition probability of sensor node state from t_i to t_{i+1} under the given control action a. Figure 4 depicts the state diagram of the

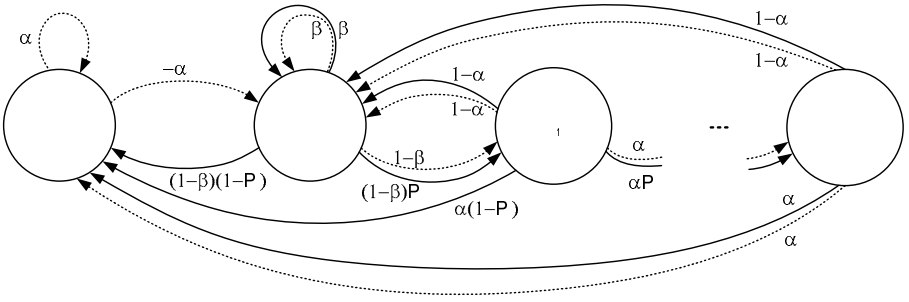


Fig. 4. State diagram of the BDT scheme. Solid line indicates the transmit action whereas dotted line does the defer action.

behavior of sensor node in which the transition probability $P_t(t_i, t_{i+1}, a)$ is given by

$$\begin{aligned}
P_t(I, I, \cdot) &= \alpha \\
P_t(I, B_0, \cdot) &= 1 - \alpha \\
P_t(B_0, B_0, \cdot) &= \beta \\
P_t(B_0, B_1, 0) &= 1 - \beta \\
P_t(B_0, B_1, 1) &= (1 - \beta) P_f(g) \\
P_t(B_0, I, 1) &= (1 - \beta)(1 - P_f(g)) \\
P_t(B_i, B_{i+1}, 0) &= \alpha, \quad 0 < i < D \\
P_t(B_i, B_{i+1}, 1) &= \alpha P_f(g), \quad 0 < i < D \\
P_t(B_i, B_0, \cdot) &= 1 - \alpha, \quad 0 < i \leq D \\
P_t(B_i, I, 1) &= \alpha(1 - P_f(g)), \quad 0 < i < D \\
P_t(B_D, I, \cdot) &= \alpha
\end{aligned}$$

Average Cost. Let $\pi = \{\mu_0, \mu_1, \mu_2 ...\}$ be a policy which maps system states into control actions. Our objective is to minimize over all policies π with $\mu_t : S \rightarrow A, \mu_i(s) \in A(i)$ for i and s, the average cost per stage

$$J_\pi(s) = \lim_{T \rightarrow \infty} \frac{1}{T} \left\{ \sum_{i=0}^{T-1} C(s_i, \mu_i(s_i)) \mid s_0 = s \right\} \tag{6}$$

We need to choose an optimal policy π to minimize this cost:

$$\pi^* = \arg\min_{\pi} J_\pi(s) \tag{7}$$

Our model is an unichain MDP model in which its stationary policy and optimal average cost can be obtained by using the relative value iteration algorithm in the dynamic programming techniques [8][14].

4 Numerical Results and Simulations

In this section we analyze the performance of our transmission scheme named BDT under various operating conditions. Extensive simulations are also performed with ns-2 simulator to verify energy efficiency and throughput of our scheme under delay constraints. We consider 30 sensor nodes randomly dispersed over a network field (350m $\times$ 350m) and 1 sink node. Only 15 sensor nodes generate a frame of 128 bits destined to the sink node and the rest of the sensor nodes only relay the frames to the sink node. The on-off traffic model is used to generate the traffic. Wireless fading channel is modeled by 20-state Markov chain model. Table 1 summarizes the values of the various parameters used in our experiments. Note that T_f in our simulation is configured to match the period in which sending and receiving nodes complete the exchange of RTS, CTS, DATA, and ACK frames using RTS/CTS mechanism. Besides the additive parts of 802.11 such as IP header and MAC header are tripped off in our simulations.

Table 1. Parameter values used in the simulations and numerical results

Parameters	Value (default)
Number of sensor nodes	30
Mobility of sensor nodes	Stationary
Frame transmission time (T_f)	1 ms
Transmission range of sensor node	150 m
Source rate during "ON" state	128 Kbps
Routing protocol	DSDV
Doppler frequency	10 Hz
On-off source traffic parameters α	0.95
On-off source traffic parameters β	0.25
Lifetime of the frame (D)	$100 \cdot T_f$
Weighting factor in cost function (δ)	0.1

4.1 Optimum Thresholds for BDT

We first analyze how the optimum threshold for transmit action varies under different channel conditions. Figure 5 shows the optimum transmit threshold in dB of BDT, for varying Doppler frequency (f_m) and frame lifetime (D). As shown in the figure the optimum values of state B_i tends to increase as Doppler frequency becomes larger. This result indicates that deferring the transmission is more favorable in a fast varying channel. In a slow fading channel, deferring the transmission is not beneficial since the channel will not become better in the near future when the node currently sees bad channel. Thus the threshold should be set to be lower. However, a higher threshold is preferred under fast

fading channel to prioritize the defer decision. Another interesting observation from Fig. 5 is the thresholds for B_i, $0 < i \leq D$, are higher than B_0 for lower Doppler frequencies. That is because B_0 is in "ON" state in which traffic is generated, thus faster transmission is preferred to avoid buffer overflow.

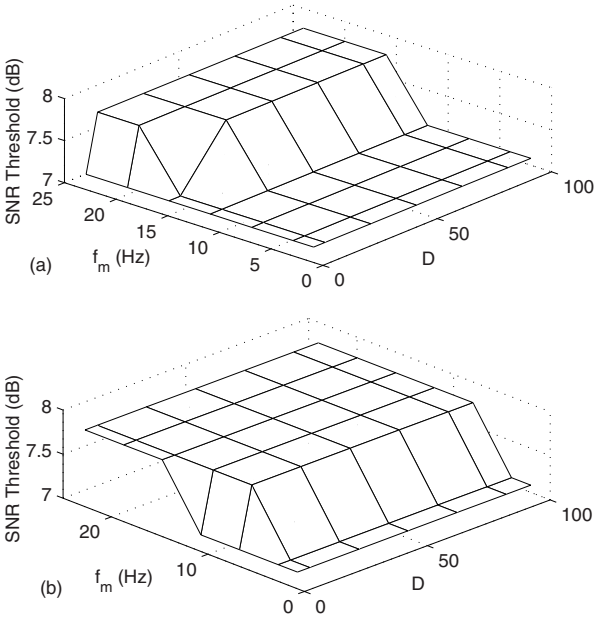


Fig. 5. The optimum thresholds of transmit action versus Doppler frequency and delay constraint ($\rho = 10$dB, $\alpha = 0.95$, $\beta = 0.25$). (a) state B_0. (b) states B_i, $0 < i \leq D$.

Figure 6 shows the impacts of traffic load on the optimum threshold for transmit action. As shown in the figure, the thresholds for B_0 decreases when the traffic load becomes heavier. Note that lower α and higher β corresponds to heavier traffic. In contrast, the thresholds for B_i, $0 < i \leq D$ are sensitive to α only. This is because the state B_i, $0 < i \leq D$ implies that source traffic is in "OFF" state, independent of the probability β of "ON"-to-"ON" transition probability.

Figure 7 shows the thresholds for transmit action under varying α and delay constraint D. As shown in the figure, the thresholds tend to increase for a fixed α when the lifetime of frame is extended from 5 to 100 slots. To be more precise, this observation seems to be true only for relatively larger values of α. As α becomes smaller, this threshold remains almost constant over $5 \leq D \leq 100$ slots. These results indicate that delay constraint significantly influences the transmit threshold when it is rather shorter than the OFF period. This is somewhat obvious because frame in the buffer will be preempted by new arriving frame when delay constraint is larger than mean OFF period. Thus delay constraint

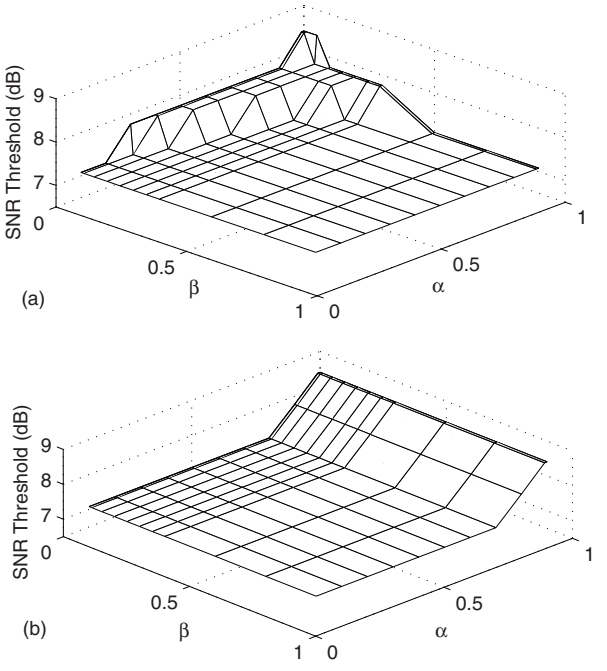


Fig. 6. The optimum thresholds for transmit action versus the traffic parameters α and β ($\rho = 10$dB, $f_m = 20$Hz, $D = 100 \cdot T_f$). (a) State B_0. (b) States B_i, $0 < i \leq D$.

becomes meaningless in this case. Figure 8 shows the dependence of the transmission threshold on weighting factor. As shown in this figure, this threshold is relatively sensitive to the weighting factor. This result is clear because of the weighting factor's role in the cost function described in equation (2).

4.2 Energy Efficiency of 802.11 with BDT Versus Plain 802.11

Figure 9 shows simulation results of energy efficiency for two transmission schemes: 802.11 with BDT and plain 802.11. Here the energy efficiency is defined as the ratio of the number of successful transmissions over the number of transmission attempts. As shown in the figure, the 802.11 with BDT and plain 802.11 has the energy efficiency of 0.96, and 0.71, respectively ($\rho = 10$dB). A significant improvement, more than 35 %, is achieved with our proposed scheme, compared with the existing equivalent counterpart. Such energy conservation mostly comes from the reduction in the number of frame loss due to transmission errors. However, such gains does not come for free. As shown in the next figure, our transmission scheme based on the opportunistic transmission sacrifices throughput by deferring the transmission when the channel state is bad.

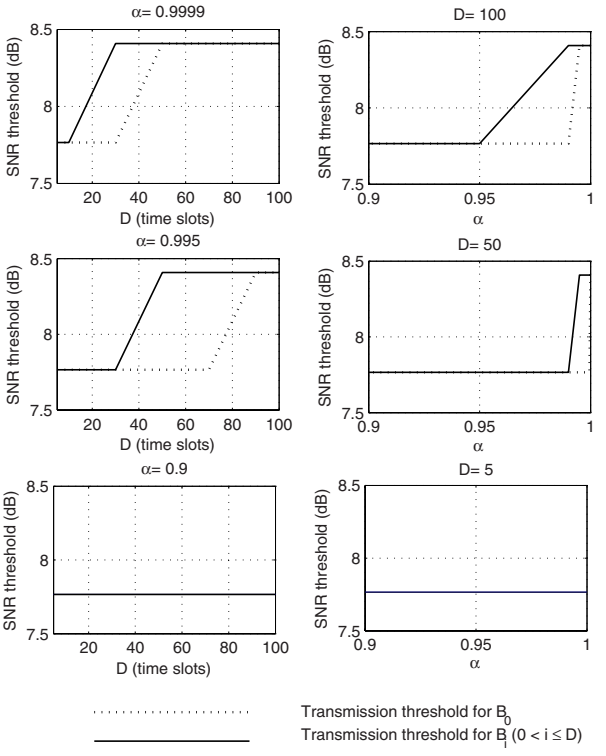


Fig. 7. The optimum thresholds of transmit action versus α and D ($\rho = 10$dB, $\delta = 0.1$, $\beta = 0.05$, $f_m = 20$Hz)

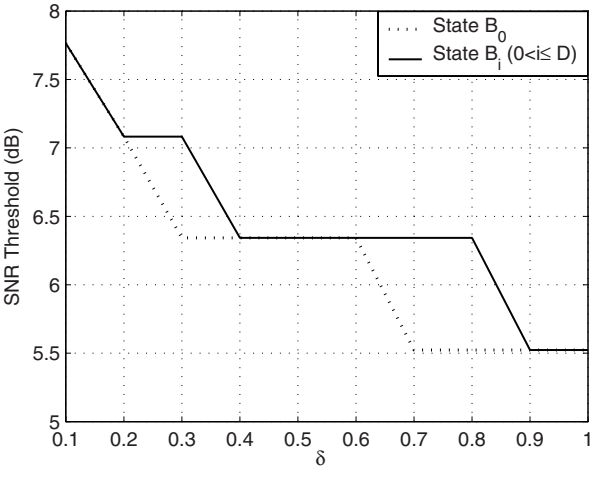


Fig. 8. The optimum thresholds of transmission versus the weighting factor δ ($\rho = 10$dB, $\alpha = 0.95$, $\beta = 0.25$, $f_m = 20$Hz, $D = 100 \cdot T_f$)

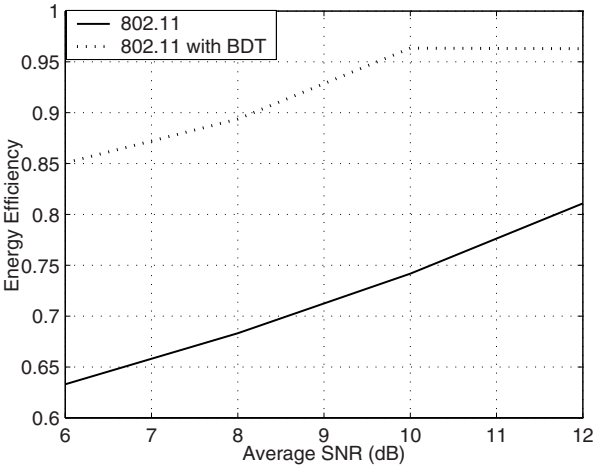


Fig. 9. Energy efficiency of two transmission schemes: 802.11 with BDT, and plain 802.11

4.3 Throughput of 802.11 with BDT Versus Plain 802.11

Figure 10 shows throughput of two alternative transmission schemes: 802.11 with BDT and plain 802.11. Throughput is observed at the sink node and averaged over the simulation time. As shown in the figure, plain 802.11 has higher throughput than 802.11 with BDT, although the gap is not significant. As mentioned

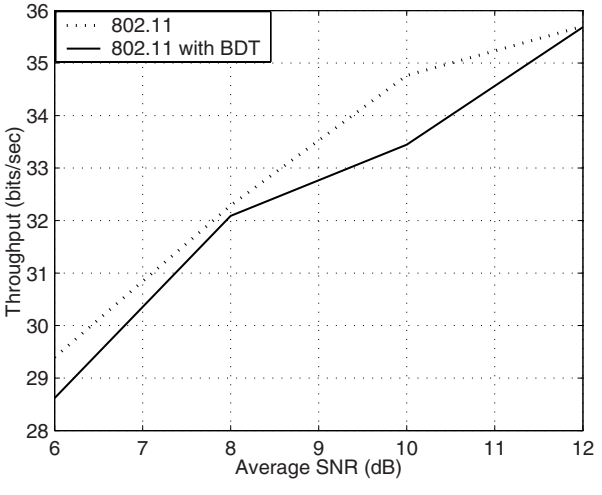


Fig. 10. Throughput of two transmission schemes: 802.11 with BDT, and plain 802.11

earlier, a higher energy efficiency can be achieved at the expense of throughput. Considering the trade-off issue, we can see throughput performance of our proposed scheme is not significantly deteriorated.

5 Conclusion

We propose an opportunistic transmission strategy for wireless sensor networks that operate in a strict energy-constrained environment. The WSNs of interest in this work are assumed to host delay-sensitive applications. Our proposed transmission strategy named BDT attempts to transmit at good channel conditions while meeting the delay constraint, under time-varying wireless channel. The Markov decision process is used to find optimum thresholds for transmit action. Our BDT scheme initiates transmission only when the channel quality exceeds the optimum threshold, so that unsuccessful transmissions causing a waste of energy are avoided whenever possible. This scheme can be applied for existing MAC protocols. Extensive simulations are performed to verify the performance of our proposal over wireless fading channels. The simulation results show that the BDT scheme has 35% higher energy efficiency than its counterparts, thereby further prolonging the network lifetime.

References

1. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Commun. Mag. 40(8), 102–114 (2002)
2. Ephremides, A.: Energy concerns in wireless networks. IEEE Wireless Commun. 9(4), 48–59 (2002)
3. Stark, W., Wang, H., Worthen, A., Lafortune, S., Teneketzis, D.: Low-energy wireless communication network design. IEEE Wireless Commun. 9(4), 60–72 (2002)
4. Min, R., Bhardwaj, M., Cho, S.H., Ickes, N., Shih, E., Sinha, A., Wang, A., Chandrakasan, A.: Energy-centric enabling technologies for wireless sensor networks. IEEE Wireless Commun. 9(4), 28–39 (2002)
5. Goldsmith, A.J., Wicker, S.B.: Design challenges for energy-constrained ad hoc wireless networks. IEEE Wireless Commun. 9(4), 8–27 (2002)
6. Ci, S., Sharif, H., Nuli, K.: Study of an adaptive frame size predictor to enhance energy conservation in wireless sensor networks. IEEE J. Select. Areas in Commun. 23(2), 282–292 (2005)
7. Wang, J., Zhai, H., Fang, Y.: Opportunistic packet scheduling and media access control for wireless LANs and multi-hop ad hoc networks. In: Proc. WCNC'04. vol. 3, pp. 1234–1239 (March 2004)
8. Hoang, A.T., Motani, M.: Buffer and channel adaptive transmission over fading channels with imperfect channel state information. In: Proc. WCNC'04. vol. 3, pp. 1891–1896 (March 2004)
9. Holland, G., Vaidya, N.H., Bahl, P.: A rate-adaptive MAC protocol for multi-hop wireless networks. In: Proc. MOBICOM 2001, pp. 236–251 (July 2001)
10. Phan, C.V., Kim, J.G.: An energy-efficient transmission strategy for wireless sensor networks. In: Proc. WCNC '07 (March 2007)

11. IEEE Std. 802.11. Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) Specifications (June 1999)
12. Puterman, M.L.: Markov decision processes: discrete schochastic dynamic progrmaming Hoboken. John Wiley & Sons, New Jersey (1998)
13. Wang, H.S., Moayeri, N.: Finite-state Markov channel-a useful model for radio communication channels. IEEE Trans. Veh. Technol. 44(1), 163–171 (1995)
14. Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction. MIT Press, Cambridge (1998)
15. Charles, C., Liu, K.J.R.: Near-optimal reinforcement learning framework for energy-aware sensor communications. IEEE J. Select. Areas in Commun. 23(4), 788–797 (2005)
16. Angel, G.D., Fine, T.L.: Optimal power and retransmission control policies for random access systems. IEEE/ACM Trans. Networking 12(6), 1156–1166 (2004)
17. Pham, P.P., Perreau, S., Jayasuriya, A.: New cross-layer design approach to ad hoc networks under Rayleigh fading. IEEE J. Select. Areas in Commun. 23(1), 28–39 (2005)
18. Chen, Y., Zhao, Q.: On the lifetime of wireless sensor networks. IEEE Commun. Lett. 9(11), 976–978 (2005)
19. Chen, Y., Zhao, Q.: Maximizing the lifetime of sensor network using local information on channel state and residual energy. In: Proc. CISS '05 (March 2005)
20. Fu, A., Modiano, E., Tsitsiklis, J.N.: Optimal transmission scheduling over a fading channel with energy and deadline constraints. IEEE Wireless Commun. 5(3), 630–641 (2006)
21. Johnston, L.A., Krishnamurthy, V.: Optimality of threshold transmission policies in Gilbert Elliott fading channels. In: Proc. ICC '03, vol. 2, pp. 1233–1237 (May 2003)
22. Zhang, D., Wasserman, K.M.: Energy efficient data communication over fading channels. In: Proc. WCNC '00, vol. 3, pp. 986–991 (September 2000)

Workflow-Level Parameter Study Support for Production Grids*

Peter Kacsuk, Zoltan Farkas, and Gabor Hermann

MTA SZTAKI, 1111 Kende utca 13
Budapest, Hungary
{kacsuk,zfarkas,gabor.hermann}@sztaki.hu

Abstract. Workflow applications are frequently used in many production Grids. There is a natural need to run the same workflow with many different parameter sets. Unfortunately current Grid portals either do not support this kind of applications or give only specialized support and hence users are obliged to do all the tedious work needed to manage such parameter study applications. P-GRADE portal has been providing a high-level, graphical workflow development and execution environment for various Grids (EGEE, UK NGS, GIN VO, OSG, TeraGrid, etc.) built on second and third generation Grid technologies (GT2, LCG-2, GT4, gLite). Feedback from the user communities of the portal showed that parameter study support is highly needed and hence the recent release of the portal supports the workflow-level parameter study applications. The current paper describes the semantics and implementation principles of managing and executing workflows as parameter studies. Two algorithms are described in detail. The black box algorithm optimizes the usage of storage resources while the PS-labeling algorithm minimizes the load of Grid processing resources. Special emphasis is on the concurrent management of large number of files and jobs in the portal and in the Grids as well as providing a user-friendly, easy-to-use graphical environment to define the workflows and monitor their parametric study execution.

1 Introduction

One of the most promising utilizations of Grid resources comes to life with parameter study (or sometimes written as "parametric study" or "parameter sweep") applications where the same application should be executed with a large set of input parameters. Such parameter study applications are easy to implement in the Grid since the different executions started with different parameters are completely independent. Indeed, there are several projects [1], [2] that demonstrated that parameter study applications are easily manageable in the Grid. However, most of these projects tackled only single job based applications. The real challenge comes when complex applications consisting of large number of jobs/services connected into a workflow

* The work reported in this paper is partly supported by the European Commission under contract numbers LSHC-CT-2006-037559 (FP6 STREP, CancerGrid, www.cancergrid.eu) and 031775 (FP6 SEE-GRID EU project, www.see-grid.eu).

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 872–885, 2007.
© Springer-Verlag Berlin Heidelberg 2007

should be executed with many different parameter sets. As far as we know there have been only two projects that tried to combine parameter studies with workflow-level support in the Grid. ILab [3], [4] enables the user to create a special parameter study oriented workflow. With the help of a sophisticated GUI, the user can explicitly define how to distribute and replicate the parameter files in the Grid and how many independent jobs are to be launched for each segment of the data files. This approach is very static restricting the exploitation of the dynamic nature of Grids that enables the dynamic collection of resources. The SEGL [5] approach puts much more emphasis on exploring the dynamic nature of the Grid. They also provide a GUI to define the workflows and to hide the low level details of the underlying Grid. The SEGL workflow provides tools for several levels of parameterization, repeated processing, data archiving, handling conclusions and branches during the processing as well as synchronization of parallel branches and processes. The problem with this GUI is that it might be too sophisticated, requiring very large skill from the application developer. Furthermore both ILab and SEGL are connected to a particular Grid although in case of a parameter study execution there is a large need to exploit as many resources as possible even if they should be collected from different Grids. Finally, neither of them can be used as a service through a Grid portal.

Although our approach to support workflow-level parameter study applications in the Grid has many similarities with these two projects, there are significant differences, too. Our main goals are as follows:

1. Keep both the workflow GUI and the parameter study support concept as simple as possible. This enables the fast and easy learning of the tool as well as its easy usage.
2. Enable to run any existing workflow with different parameter sets without modifying the structure of the workflow.
3. Manage the execution of the workflows on as many Grid resources as possible. Enable the collection of Grid resources from several Grids even if they are based on different Grid technology.
4. Enable the access of the workflow-oriented GUI and the available Grids via a single Grid entry point, i.e., via a Grid portal without installing any software on the user's machine.
5. Provide a dynamic balance between the usage of processing resources, storage resources and network resources.

Clearly, these goals differ from the main concept of the above mentioned two projects. The last goal can be found in several other parameter study projects that aim at the support of parameter study applications at the individual job level. In case of Apples/APST [6] the main emphasis is on scheduling and hence in this project the fifth goal of our project is dominant.

The starting point for our project was the P-GRADE Grid portal [7] that provides a workflow-oriented GUI as well as workflow-level interoperability between various Grids even if they are built on different Grid technologies. This means that the same portal can be connected to several different Grids and the portal manages the workflow execution among these Grids according to the users' requirement [8]. The portal even enables the parallel exploitation of the connected Grids, i.e., different jobs of the same workflow can simultaneously be executed on Grid resources taken from

different Grids. Such a multi-Grid workflow execution mechanism is a unique feature of the P-GRADE portal that is now widely used for many different Grids (SEE-GRID, VOCE, EGrid, HunGrid, CroGrid, GILDA, etc.). Besides LCG-2 and gLite based production Grids the portal is successfully used as service for the GT2 based UK National Grid Service (NGS) and it was also successfully connected to the GT4 based Westfocus Grid (UK). Recently the GIN (Grid Interoperation/Interoperability Now) VO of OGF is supported by the portal enabling the simultaneous access to all of its resources coming from different Grids including US OSG and TeraGrid, UK NGS, UK WestFocus Grid and EGEE Grids.

Experiences of the portal revealed that many applications require not only the single execution of a workflow rather users seek for parameter study support to execute an existing workflow application with many different parameter sets. Therefore, our motivation was to extend the existing single workflow support of the portal towards a generic workflow-level parameter study execution support. Such support should enable the automatic starting, execution, monitoring and visualization of all the workflows belonging to the same parameter study. Of course, the same way as in the case of the single workflow management environment, the users should neither know any details of the underlying Grids nor insist on any particular programming language. Even legacy codes can be used as services in the workflows if the portal is integrated with the GEMLCA legacy code architecture service [9].

In order to reach the five main goals mentioned above we have developed two algorithms to manage workflow-level parameter studies across multiple Grids. The first algorithm is based on the "black box" concept that minimizes the required storage resources of the portal but results in superfluous job executions in many cases. The second algorithm, called as the PS-labeling algorithm minimizes the usage of processing resources of the Grids but increases the storage needs of the portal. They represent two extreme points in the scale of possible execution management algorithms. The algorithms should be mixed according to a dynamic and adaptive scheduling algorithm that is the subject of our future research.

The paper first introduces the workflow concept of P-GRADE portal in Section 2. The next section explains the "black box" execution semantics and its portal support both at the user interface and the portal workflow manager level. Section 4 introduces the PS-labeling algorithm and compares it with the "black box" execution mechanism. Finally, Section 5 compares our research with related ones.

2 The Workflow Concept of P-GRADE Portal

When designing and implementing P-GRADE portal, we focused on the following basic principles.

Simplicity and expressiveness of the user interface
P-GRADE portal provides a graphical editor to develop workflow applications. The workflow graph is a simple DAG (Directed Acyclic Graph) where nodes of the graph are either jobs (sequential, MPI or PVM) or legacy code services (if GEMLCA is integrated with the portal). For each node input and output ports can be defined. Input ports represent the input files to be used by the node and output ports represent output files to be generated by the node. Input and output ports can be connected by directed

arcs (from the input to the output port). The arcs represent the necessary file transfer between the two connected nodes. Indeed, the portal run-time system automatically takes care of these predefined file transfers, so the user is completely released of transferring files among jobs during the workflow execution. A node can be executed if all the input arcs have received the necessary input files. This very simple dataflow semantics enable the parallel execution of those nodes that are simultaneously supplied with the required input files. Notice that based on this semantics the user can easily achieve two-level parallelism. The first level is among nodes of parallel branches of the workflow (inter-node parallelism), the second level is within a node if the job executable defined for the node is a parallel code (MPI or PVM). This second level is called intra-node parallelism. To keep the concept as simple as possible neither if-then-else nor loop constructs are provided in the graph.

In order to illustrate the graphical workflow concept of P-GRADE portal Figure 1 shows a simple example workflow that is used for solving the Ax = B equation where A and B are matrices, and x is a vector. The application consists of 5 jobs (all of them having sequential executable code). The first job called as "Separator" accepts the A and B matrices as input parameters on its input port, separates them and then copies A to jobs "Invert_A" and "A_mul_X", and copies B to "Multip_B" and "Subtr-B". The job "Invert_A" creates the invert matrix of A that is multiplied by B in job "Multip_B". The output of "Multip_B" is the searched X vector. The next two jobs are used to check the quality of the result.

Exploiting the available services of existing Grids

Instead of developing a new kind of middleware as it has been done in several other parameter study related Grid projects [6] our aim was to exploit the available services of existing production Grids. For this reason we have not developed our own enactment service rather we use Condor DAGMan. If there are alternative services, we use them in order to ensure Grid interoperability through the portal. For example, the portal can access, process and visualize the information in both MDS and BDII type information systems. However, the portal should work even if a specific service is not available. For example, the portal can exploit the EGEE broker services to assign jobs to Grid resources but it also can manage GT2 Grids where such broker does not exist (e.g. the UK NGS). In the former case the portal automatically generates a default JDL file based on the user created workflow definition and enables the user to tune this JDL file with additional visual editing. In the latter case, the portal enables the user explicitly assigning jobs to Grid resources. In both cases, first the user should select the Grid where the job should be executed. For different jobs of the workflow the user can select different Grids either with broker or without broker.

Tailoring the Grids to user needs

The portal administrator can connect the portal to several Grids. The portal maintains a list of connected Grids common for every users. This list cannot be modified by the users only by the portal administrator. For each connected Grid a list of available resources is provided for every single user. The content of these Grid resource lists can be modified by the user. Grids usually have many unreliable sites and the user can remove from his Grid resource list those sites that he identifies as unreliable ones. On the way round, if he knows some extra sites not shown by the Grid resource list he

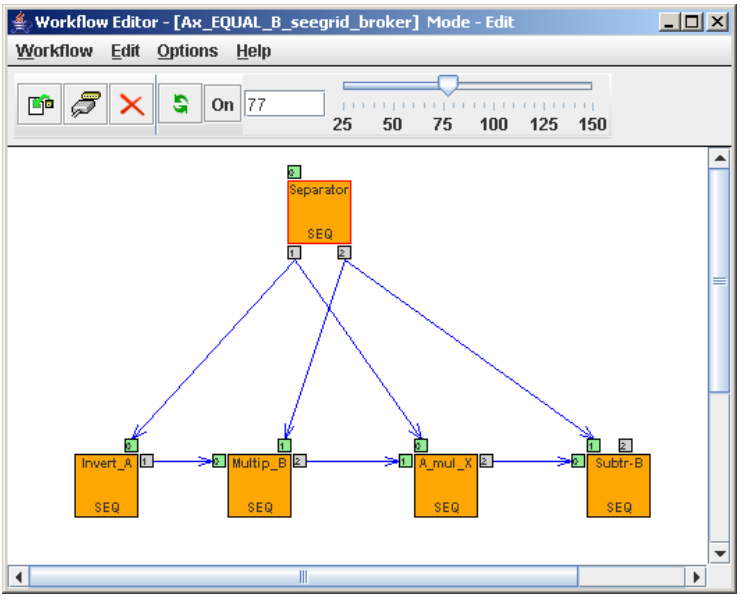


Fig. 1. Workflow for Computing the Ax=B equation

can easily add those sites to his own Grid resource list. If later the user would like to assign a job to a Grid resource, the portal offers the available Grid resources according to these Grid resource lists.

Flexibility, expandability, standardization
In order to provide a flexible, easy to expand and standard portal we selected the Gridsphere portal framework technology [15] and used it to build our portal. Gridsphere is used by a large community enabling the exchange of available portlets that are written according to the Gridsphere standard. It also enables the easy tailoring of the portal to specific user needs. Our aim with the P-GRADE portal was to create a workflow-level core portal that can easily be extended and adapted for different user communities' needs.

3 The "Black Box" Execution Semantics of Workflow-Level Parameter Study Execution

The simplest approach of supporting parameter studies at the workflow level is based on the "black box" execution semantics. It means that we consider a workflow as a black box that should be executed with many different parameter sets. These parameter sets are placed on the so-called PS input ports of the workflow. An input port is called PS input port if a set of parameter files can be received on that port. If a workflow has one such PS input port, it should be executed as many times as many elements are in the parameter file set of that port. If there are several PS input ports, the workflow should be executed according to the cross-product of these input sets.

From now on we say that a workflow (WF) which has got at least one PS input port is called a PS workflow (PS-WF).

The concept of the workflow-level PS support based on the "black box" execution semantics is illustrated in Figure 2. The original WF consisting of 4 jobs is considered as a black box that have two input ports capable of accepting inputs from the outside world. On both inputs the user can provide several input sets and the workflow should be executed with the cross-product of these input sets. Figure 2 shows a case when Input port 0 accepts an input set with two elements (value 1 and 2) and Input port 1 accepts an input set with three elements (value 3, 4 and 5). Therefore, the workflow should be executed six times as shown in Figure 2.

In order to manage the execution of workflows according to the "black box" execution semantics the workflow manager of the portal was extended in the following way. Let

$$M = N_1 \text{ x } N_2 \text{ x} \ldots \text{x } N_m$$

where *m* is the number of PS input ports and N_i denotes the number of input files on the i-th PS input port. At run-time the portal PS-WF manager generates *M* executable workflows (e-WFs) from the original PS-WF. Every e-WF is labeled by *m* labels:

$$1 - n_1, 2 - n_2 , \ \ldots \ m - n_m$$

Where the internal structure of label^i is:

$$i - n_i$$

where i identifies the i-th input PS port and n_i represents the ordering number of the input file taken from the i-th PS port in the identified execution instance:

$$0 \leq n_i < N_i$$

Figure 2 also shows these labels for each of the 6 execution instances. This labeling scheme identifies for the PS-WF manager which input file to take from the different PS input ports in the case of different execution instances (e-WFs). It also helps in identifying the output files generated at the output ports. Every output file is labeled with the label of the e-WF that generated it. Notice that output files can be local and remote. Remote output files are always permanent and once they are produced by an e-WF they can be immediately read by the user. This enables the user to study the partial results even if an e-WF is not completed. Local output files can be permanent or volatile. Permanent file means that the user would like to get access to this output file only when the whole e-WF execution has been completed. These partial results are collected and stored by the portal meanwhile the e-WF runs. When the e-WF is completed these files are zipped (together with the standard output and error logs) and placed by the portal to a Grid storage resource that was defined by the user. Finally, local volatile files represent temporary partial results. As they are consumed by the connected job(s) they can be removed by the portal.

Developing and running PS-WF applications according to the "black box" execution semantics require three main steps. The first step is the development of the WF application that the user would like to run as a PS-WF application. The development process of a WF application has been described in previous publications

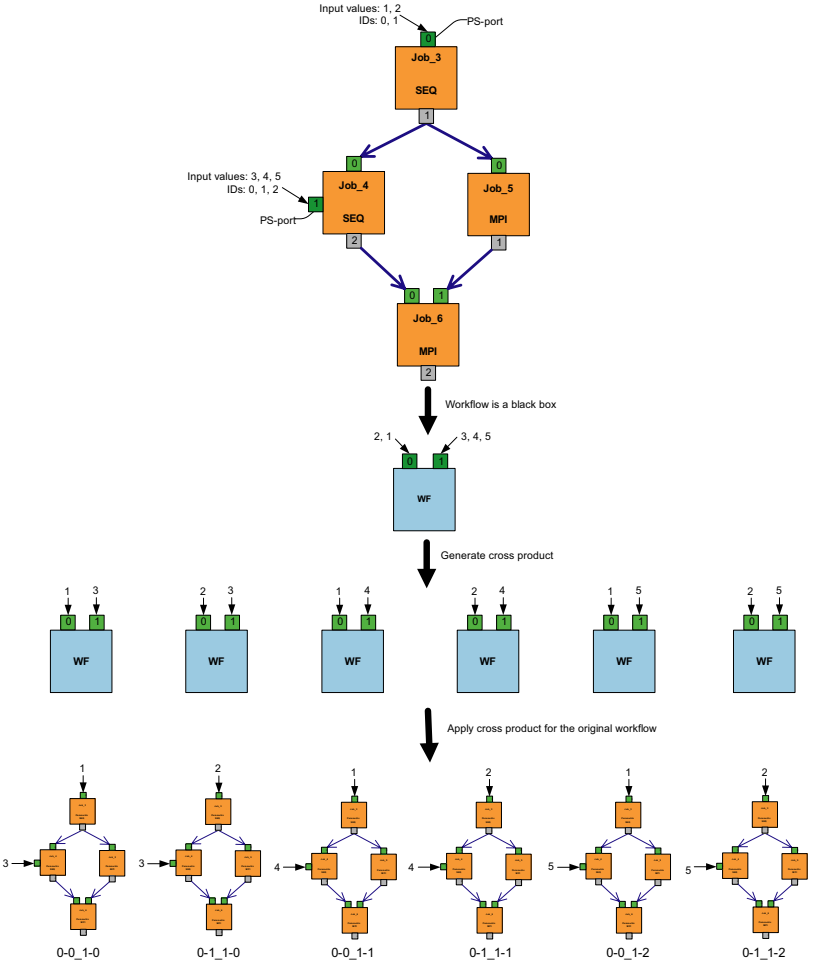


Fig. 2. Concept of "black box" execution semantics

[7] as well as in the User's Manual of the current service portals and hence we do not describe it in this paper. The basics concepts are summarized in Section 2. The second step is the transformation of the WF into a PS-WF. Finally, the third step includes the submission of the PS-WF application to the Grid and monitoring the execution of the PS-WF application. Consequently, the PS-WF user interface has two major parts:

1. Definition of PS-WF graphs
2. Monitoring the PS-WF graph execution

3.1 PS-WF Graph Definition

In order to turn a WF application into a PS-WF application the graphical Workflow Editor (WE) of the portal was slightly extended. The user can open the existing WF

by the WE and can turn any of the existing input port into a PS input port. In order to illustrate the process we use the same Ax_EQUAL_B workflow application that was shown in Section 2. The task is to modify this WF in order to solve the equation for a set of A and B parameters. Figure 3 shows how to turn the input port of the Separator job into a PS port. Notice the difference between the input port and PS input port definition. In case of a normal input port a file is associated with the port. This file can be either local (originating from the user's machine and part of the input sandbox) or remote (placed in a storage resource of the Grid). In case of a PS port (Figure 3) a directory is associated with the port. This directory always should be placed in a storage resource of the Grid. The user should place the series of input files into this directory that must not contain any other file. After defining the PS input ports the user should identify the Grid and its storage resource where the local permanent files should be stored at the end of each e-WF execution. As a summary we can say that turning an existing WF into a PS-WF is an extremely easy task. Simply turn some of the input ports to PS input ports and define the target Grid storage resource for the local permanent files.

3.2 Monitoring the PS-WF Graph Execution

Even monitoring a single job is important for the user not mentioning when he runs thousands of jobs as part of a PS-WF execution. The challenge here is how to visualize the execution status of thousands of e-WFs and jobs in an easily understandable and manageable way. Again the monitoring of a single WF as it was managed by the portal was a good starting point. It was only slightly extended and yet it can help to monitor all the e-WFs and all their jobs in an efficient way. The ordinary WFs of a user are listed by the portal under the Workflow Manager window. Here the user can submit the WF, attach the WF to the Workflow Editor to see the graphical view of the WF, and delete the WF. Moreover, the "Details" button enables the user to see the details of the WF, i.e., to see the component jobs and their assignment to Grid resources. The PS-WFs are listed in the Workflow Manager window in the same way as the WFs. The only difference is that the PS-WFs have a "PS Details" button to show their internal details.

Once the user submits the PS-WF (simply clicking on the corresponding "Submit" button), the portal workflow manager (WM) creates all the e-WFs that are defined by the cross-product of the PS input ports' file sets. Then WM submits simultaneously as many e-WFs as many are permitted by the portal administrator. In principle all the e-WFs could be submitted into the Grid. In order to prevent the flooding of the portal and the Grid by the e-WFs and jobs of a single user, the portal administrator can restrict the number of e-WFs that can simultaneously be submitted to the Grid. After the PS-WF submission using the PS Details button the user can see the statistics of the e-WFs: how many were initiated, submitted, finished and how many went on Rescue or Error. Figure 4 shows the situation where the "Ax_EQUAL_B_PS" PS-WF was submitted with 6 input parameter sets. As a result 6 e-WFs were generated by the portal. 2 of them already finished, 3 submitted and one still in init state, i.e. waiting

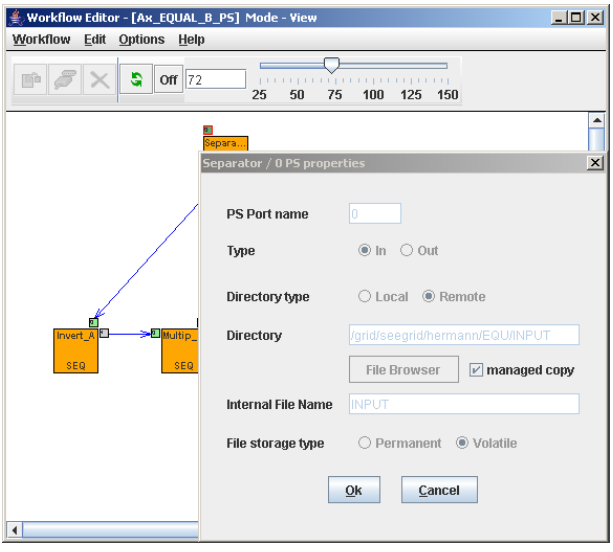


Fig. 3. Definition of a PS input port

for submission. The figure also shows that any submitted e-WF can be viewed in detail by using the "Details" button. Clicking there the detailed view of the e-WF shows the component jobs of the e-WF, their Grid resource assignment and their status. Notice that any e-WF can be aborted. It kills the selected e-WF but the other e-WFs can continue their activity. Figure 4 also reveals that those e-WFs that are finished cannot be viewed in the PS Workflow Details window. Their results (including every stdout and stderr files) are already stored in the defined Grid storage resource so the user can check those files there.

4 PS-Labeling Algorithm

Although the "black box" execution semantics is easy to understand and use, unfortunately it does not represent an optimal execution semantics. In the case of the example shown in Figure 2 the whole PS-WF as a black box should be executed 2x3=6 times, i.e., all the jobs of the PS-WF will be executed 6 times. However, analyzing the PS-WF it quickly turns out that it is enough to execute job3 and job5 only with the two input values of PS input port 0. On the contrary, Job4 and Job6 indeed should be executed with the cross-product of the two PS input ports, i.e. 6 times. This example shows that the "black box" execution semantics results in a redundant execution that is not tolerable if the number of files on the PS input ports are in the range of hundreds and thousands as it frequently happens during large-scale scientific simulations.

In order to be able to solve the problem illustrated by Figure 2 we introduce a new execution mechanism called as "PS-labeling". The new method is based on the "black box" approach in the sense that the user should not have to modify anything related to an existing WF, except for simply turning the required input ports into PS input ports.

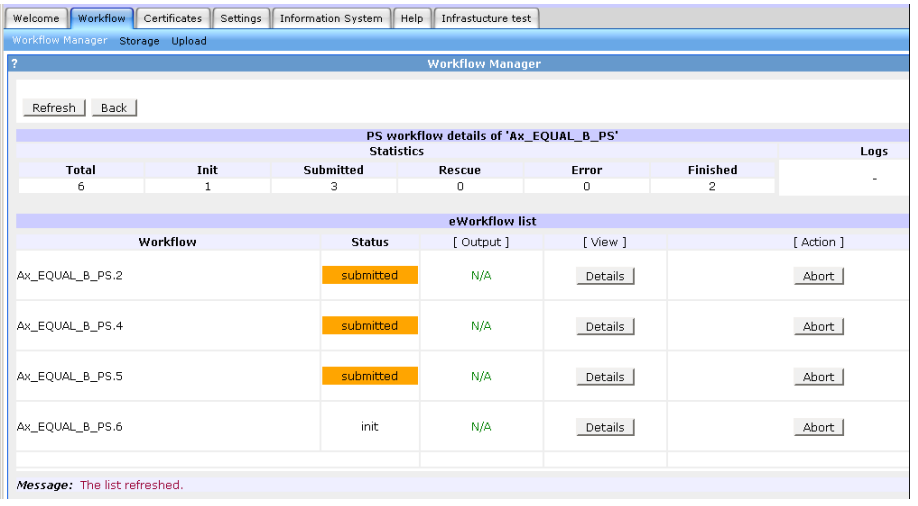


Fig. 4. Detailed monitoring view of a PS-WF's execution

Once this is done the portal will handle the WF as a PS-WF and provides a unique natural number as PS port identifier for the PS input ports. Of course, the user should place the necessary number of input files for every PS input port. Based on the PS port identifier and the number of input files placed on the PS ports, the portal can identify for each node of the PS-WF graph the optimal execution number. This is done by the PS-labeling algorithm run by the workflow manager of the portal. The PS-labeling algorithm has two phases:

- Preparation phase
- Execution phase

Preparation phase

The preparation phase is executed by the workflow manager before submitting a PS-WF. Starting from each node having a PS input port represented by the label i-N_i (where i denotes a unique natural number index of the PS port and N_i denotes the number of input files belonging to this port) the algorithms extends the current label of all dependent nodes by i-N_i. (A node j is dependent from node i, if there is a directed path of arcs starting from node i and ending at node j.)

At the end of the algorithm every node is either without any label or labeled with a series of labels:

$$label^1, label^2, \ldots label^n$$

The former case means that the node does not depend on any PS input ports and hence it should be executed only once. The latter case means that the node was on at least n paths that are rooted from n different input PS ports' node. Overall such a node should be executed

$$N_1 \times N_2 \times \ldots \times N_n$$

times as the cross-product of the initial input file sets. This label set is extended to be a full label set for every node. A full label set is an ordered set of the N_i values, i.e.,

on position i there is the value of N_i. If there was no label for position j (PS port j) in the generated label set, then an empty position is placed here. The full label set for a given node shows those PS input ports that have impact on this node and for such PS ports it gives the number of input files demonstrating the strength of this impact. Example: let us suppose that M=4, N_0= 4, N_1=3, N_2=5 and N_3=2 and the generated label set for node ***A*** dependent on PS port 1 and PS port 3 is: 1-3, 3-2. Then the full label set for ***A*** will be: {_, 3, _, 2}.

After completing the labeling, for each node a power set of the full label set is generated. For node ***A*** in the example above the power set will look like:

{_,1,_,1; _,1,_,2; _,2,_,1; _,2,_2; _,3,_,1; _,3,_,2 }

Execution phase

After completing the preparation phase, the workflow manager begins to generate and submit the e-WFs labeled as

$n_1, n_2, \ldots n_m$

where i identifies the i-th input PS port and n_i represent the ordering number of the input file taken from the i-th PS port in the identified execution instance:

$0 \leq n_i < N_i$

During the e-WF execution the workflow manager tries to substitute the execution of the node by the file(s) resulting from a previous run of the node. First the e-WF label is matched with the power set of the nodes of the PS-WF. If there is no power set of a node it means that it has to be executed only once and marked as executed for later e-WFs. If a node has a power set and the matching element is marked for the node, then this node was already executed and therefore the workflow manager does not submit the job (node) again, rather immediately provides those output files that were generated by this node during the marked run. If the matching element is not marked for a node, then this node should be executed. After executing the node its matching element is marked and the result output files are stored either in a Grid storage resource if they were defined as remote files or inside the portal if they were defined as local files.

To illustrate the algorithm let us take the example shown at the preparation phase (M=4, N_0= 4, N_1=3, N_2=5 and N_3=2) and consider the e-WF with label “3,2,4,1”. If node ***A*** has the power set {_,1,_,1; _,1,_,2; _,2,_,1; _,2,_2; _,3,_,1; _,3,_,2 } and the matching element _,2,_,1 is not marked yet, the job or service belonging to node ***A*** should be executed. Once it is executed the matching element _,2,_,1 should be marked. If later another e-WF for example with label “1,2,3,1” is to be executed, then the workflow manager will recognize that the matching element _,2,_,1 is already marked and hence node ***A*** must not be executed again.

As a conclusion we can say that the more internal output files are defined as remote files the less storage price we have to pay for the optimized usage of processing resources. Since remote files also help in on-line checking and monitoring the execution of PS-WFs, it is recommended to use remote files instead of local files in case of PS-WFs. On the other hand remote files restrict the distribution of the

execution of PS-WF nodes in different Grids. Usually, a node should be executed in the same Grid where the associated input/output remote file is stored. However, since local files are stored on the portal their processing PS-WF nodes can be assigned to any Grid connected to the portal (provided that the user has certificate and VO membership for the given Grid and VO). Therefore if a user would like to exploit many Grids for the execution of a PS-WF it is recommended to use local files instead of remote files. In this case the choice of Grids and Grid resources can be done on-the-fly by the portal provided that it is supported by a meta-broker. To develop the optimal Grid selection algorithm and developing a meta-broker is the subject of future research [10].

5 Related Research

There are two main streams of research directions that have strong relationship with our work. Research on scientific workflows in Grid environments is an exciting and richly investigated subject. A good survey in this field has been written by Yu and Buyya [11] and a whole special issue was recently devoted to this subject in Journal of Grid Computing [12]. These papers clearly show that meanwhile significant efforts are made to increase the usability of scientific workflows in Grid environment from many aspects (workflow composition, scheduling, performance estimation, fault-tolerance, etc.) there was little work on how to define and manage parameter studies at the workflow level.

The second main direction relevant to our investigation is the research on parameter studies. Some of the existing tools like Condor [1], UNICORE [13] or AppLeS (Application-Level scheduler) [2] can be used to launch pre-existing parameter studies within distributed resources. There are several important projects specifically aiming at the realization of parameter studies in Grid environments but most of them consider only individual jobs and not workflows [6], [14]. Usually, their main concern is the scheduling aspects of jobs in Grids. In many cases they define specialized middleware in order to optimize the scheduling [6] or introduce new scheduling concepts.

There are very few projects that deal with the integration of parameter study and workflow research. VisPortal supports workflow-level parameter study applications but it is not a general purpose portal rather a specialized one to support only rendering applications. It was based on the Grid Portal Development Toolkit that is the predecessor of GridSphere used for building P-GRADE portal. ILab [3] and SEGL [5] are two main projects that provide a graphical workflow concept particularly tailored to support large parameter study applications. Notice that their goal is different from our goal. They want to support the parameter study applications by a workflow whose components are used to specify the next stage of parameter study processing activity. As a result their workflow is much more sophisticated than the P-GRADE workflow and its creation requires much more skill. Our goal was that existing DAG workflows should be enabled to be executed as parameter studies simultaneously exploiting as many Grids and Grid resources as possible.

6 Conclusions

We have described two algorithms for executing PS-WFs in a multi-Grid environment. The black box algorithm gives an optimal solution concerning the utilization of storage resources while the PS-labeling algorithm optimizes the utilization of processing resources. Further research is required to create a dynamic and adaptive integration of the two methods where the portal can dynamically decide which method to be used for a particular node of the PS-WF graph. The current execution method of PS-WFs enables the static distribution of nodes between different Grids and different Grid resources if brokers are not available in the connected Grids. If brokers are available Grid resources can be assigned dynamically but the Grid assignment is still static. To provide a fully dynamic allocation of Grids and Grid resources a meta-broker should be developed and connected to the portal and to the Grids. The development of such a broker is subject of further research in the framework of the EU CoreGrid project.

References

1. Thain, D., Tannenbaum, T., Livny, M.: Condor and the Grid. In: Berman, F., Hey, A.J.G., Fox, G. (eds.) Grid Computing: Making The Global Infrastructure a Reality, pp. 299–336. John Wiley, Chichester (2003)
2. Casanova, H., Obertelli, G., Berman, F., Wolski, R.: The AppLeS Parameter Sweep Template: User-Level Middleware for the Grid. In: Proceedings of the Super Computing (SC 2002) Conference, Dallas / USA (2002)
3. Yarrow, M., McCann, K., Biswas, R., van der Wijngaart, R.: An Advanced User Interface Approach for Complex Parameter Study Process Specification on the Information Power Grid. In: Parashar, M. (ed.) GRID 2002. LNCS, vol. 2536, Springer, Heidelberg (2002)
4. McCann, K. M., Yarrow, M., deVivo, A., Mehrotra P.: ScyFlow: An Environment for the Visual Specification and Execution of Scientific Workflows. In: GGF10 Workshop on Workflow in Grid Systems, Berlin (2004)
5. Currle-Linde, N., Boes, F., Lindner, P., Pleiss, J., Resch, M.M.: A Management System for Complex Parameter Studies and Experiments in Grid Computing. In: Gonzales, T. (ed.) Proc. of the 16th IASTED Intl. Conf. on PDCS, Acta Press (2004)
6. Casanova, H., Berman, F.: Parameter Sweeps on the Grid with APST. In: Berman, F., Hey, A.J.G., Fox, G. (eds.) Grid Computing: Making The Global Infrastructure a Reality, pp. 773–788. John Wiley, Chichester (2003)
7. Kacsuk, P., Sipos, G.: Multi-Grid, Multi-User Workflows in the P-GRADE Grid Portal. Journal of Grid. Computing 3(3-4), 221–238 (2005)
8. Kacsuk, P., Kiss, T., Sipos, G.: Solving the Grid Interoperability Problem by P-GRADE Portal at Workflow Level. In: Proc. of the Grid-Enabling Legacy Applications and Supporting End User Workshop, in conjunction with HPDC'06, Paris, pp. 3–7 (2005)
9. Delaitre, T., et al.: GEMLCA: Running Legacy Code Applications as Grid Services. Journal of Grid. Computing 3(1-2), 75–90 (2005)
10. Kertesz, A., Kacsuk, P.: Grid Meta-Broker Architecture: Towards an Interoperable Grid Resource Brokering Service. In: CoreGRID Workshop on Grid Middleware in Conjunction with EuroPar'06, Dresden (2006)

11. Yu, J., Buyya, R.: Taxanomy of Workflow Management Systems for Grid Computing. Journal of Grid. Computing 3(3-4), 171–200 (2005)
12. Deelman, E., Taylor, I.: Special Issue on Scientific Workflows in Grid Environments. Journal of Grid. Computing 3(3-4), 151–304 (2005)
13. Erwin, D. (ed.), Joint Project Report for the BMBF Project UNICORE Plus Grant Number: 01 IR 001 A-D, Duration (January 2000 to December 2002) ISBN 3-00-011592-7
14. Abramson, D, Lewis, A., Peachy, T.: Nimrod/O: A Tool for Automatic Design Optimization. In: The 4th International Conference on Algorithms & Architectures for Parallel Processing (ICA3PP 2000), Hong Kong (2000)
15. http://www.gridsphere.org/

Certificate Issuing Using Proxy and Threshold Signatures in Self-initialized Ad Hoc Network*

Jeonil Kang, DaeHun Nyang**, Abedelaziz Mohaisen, Young-Geun Choi, and KoonSoon Kim

Information Security Research Laboratory,
INHA University, Republic of Korea
dreamx@seclab.inha.ac.kr, nyang@inha.ac.kr
{asm, choizak, soony}@seclab.inha.ac.kr

Abstract. In ad hoc network, it is very crucial to issue certificates safely in the self-initialized scheme where the system authority exists only at the beginning of the network operation. In order to solve this problem, early studies have presented some suggestions by removing the system authority itself and using certificate chain, or by making nodes act as system authorities for issuing other nodes' certificates. In this paper, using proxy and threshold signatures, we introduce a certificate issuing scheme that can solve many problems in the previously proposed solutions. We demonstrate our scheme's performance through the simulation results. Also, we discuss the security value of our scheme's in various aspects.

1 Introduction

In ad hoc network, certificates of public keys have been a general solution for key distribution and authentication. However, single point of failure and bottleneck problems which are caused by single certificate authority makes it hard to apply certificate-based public key cryptography to ad hoc network. To solve these problems, many researches have been performed [1,2,3]. One of the solutions is self-initialized (a.k.a. self-organized) certificate management scheme in which each node generates its own public and private keys. Due to the individual generation of keys, the central node which has to issue the certificates for other nodes can distribute its burden to other nodes. For the authority itself, there are two types of configuration. In the first configuration, the authority never exists all through the operation of the network while in the second configuration it only exists at the beginning of the network operation. For the first configuration, it is required to provide an additional method to verify other certificates because these certificates are independently issued by different nodes and signed by different private keys. The researches of Capkun *et al.* [1] and Li *et al.*[2] are well-know in handling this configuration. For the second case where the system authority exists only at the beginning of the network, it does not require much cost for verifying certificates

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)(IITA-2006-C1090-0603-0028).

** Corresponding Author

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 886–899, 2007.
© Springer-Verlag Berlin Heidelberg 2007

because all certificates are singed by a single private key. However, the problem that a node can sign certificates after the beginning still remains. To solve this, the *diffusion wave* method as by Luo *et al.* [3] has been considered.

In the diffusion wave method, the node which gathers several partial certificates and makes a complete certificate gains the authority for issuing other partial certificates. Therefore, the population of nodes that have the authority increases gradually. However, the diffusion wave may have another problem named the *originality of authority* in which a misbehaving node may obtain a valid certificate legally. The originality of authority problem may also lead to another problem in which a misbehaving node moves to other place in the network and misuse the network resources or mislead other nodes using its own valid certificate. Also, from the complexity perspectives, the diffusion wave requires much network traffic because some single nodes should gather several partial certificates and find other nodes for one certificate.

In this paper, we consider the complexity of the diffusion wave based scheme. To reduce the former work's complexity represented in the communication overhead, we introduce a novel certificate issuing scheme based on the proxy and threshold signatures. In addition to the communication reduction, our scheme provides a breaking-through for the required time for the signing operation.

2 Related Works: Self-initialized Certificate Management

2.1 Self-initialization

In the self-initialization network scenario, a node in the network generates its private and public key pair. To use the self-generated key pair, a node must ensure that other nodes' private keys are not modified by a third entities. On other side, even though two nodes are in one hop neighborhood, they cannot be perfectly sure of that in application layer unless they have additional protocols. Generally, certificates are used for informing the public key which is used when a node confirms the integrity of the message signed by the corresponding private key. However, the certificates must be signed by a trusted private key.

2.2 Trusted Certificate Chain

If there are not any system authority in the network and the certificates are individually signed by different private keys, a node must have the way or simply the criteria upon which it trusts other public keys. Unfortunately, on the side of the node, nothing can be trusted but its own private and public key pair. However, once a certificate issuing is performed, the issuer himself trust the public key associated with the issued certificate. In addition, the issuer can trust other certificates that can be verified by the trusted public key. In that way, a node can make trusted certificate chain started from its own keys as rewards for issuing other nodes' certificates.

To find out a trusted certificate chain between two nodes A and B which both have their own chains, Capkun *et al.* suggested a method in which the resulting chain of concatenating two sub-chains beginning from an intersecting point of the two chains and

ending by the two nodes' corresponding certificates (i.e., certificate of A and certificate of B respectively) is used [1]. In this method, however, each node should have a number of certificates for making sub-chain. Also, even each node has that number of certificates, this does not ensure the existence of the trusted certificate chain.

In another method suggested by Li *et al.*, to establish the certificate chain, the routing path is used to forward the certificate request message using the resulting routing path if a node issues certificate for all one-hop neighbors. If a node sends the certificate request message, all nodes on backward route add their certificates in the response message sequentially. As a node, it is enough to store the neighbors' certificates only. However, this increases the total network overhead even under usual communication pattern in local area [4].

2.3 Certificate Signed by System Private Key

If all certificates are signed by the system authority's private key, all certificates are verifiable by the system authority's public key that all nodes know. However, it is hard to expect that the authority is always in the same coverage area because of the nodes' mobility. Also, if an attacker succeeds in cracking the authority, the whole system will collapse. Therefore, it is better that the system authority's private key should be fractionized and distributed to several nodes while no node possesses the whole private key. To obtain a complete certificate, a node should gather several partial certificates signed by partial private keys leading to that a few nodes that have a partial private key suffers from lots of service requests. Also, a node can be delayed from obtaining a complete certificate due to the deficiency of the nodes that have partial private keys.

Luo *et al.* introduce a method named the diffusion wave (illustrated in Fig. 1(b)) to reduce the time delay for obtaining a complete certificate. In their work, they enable the node that gains a complete certificate to have the authority for issuing a partial certificate. By doing so, the burden of a few nodes can be decentralized to many other nodes. However, this cannot reduce the overall network traffic. Also, an attacker can meet nodes which have a partial private key in the neighborhood and can try attacking them just after small movement. In addition, the diffusion wave should have strict rules for authenticating nodes. As time goes on, the originality of authority will be degraded so that we cannot expect those strict rules to apply to the nodes which newly join in the network.

3 Proposed Certificate Issuing Scheme

3.1 Preliminaries

Secret Sharing and Threshold Signature: A secret sharing is a cryptographic scheme in which a secret of an entity is distributed as partial secrets among other entities [5]. If many partial secrets over threshold are gathered, the secret can be recovered. Once the a secret is recovered, it is not secret anymore leading to that the secret owner (i.e. the corresponding entity) need to re-make and re-distribute the secret shares among other entities. However, the required operations upon secret recovery are resources' consuming ones. Therefore, for using the secret without recovery, threshold cryptography are suggested.

In the threshold signature scheme, we can obtain a signature of a document by gathering partial signatures over threshold value [6]. This is possible when the signer need to sign a document while revealing the overall secret is undesirable. In this paper, we use the threshold signature scheme for assigning the proxy signature keys to the proxy signers.

Proxy Signature: A proxy signature scheme consists of an original signer, proxy signers, and verifiers [7]. The original signer commits his warrant to sign to proxy signers. According to who can make the same signature as the original one, there are two types of proxy signature schemes which are *proxy-unprotected* and *proxy-protected*. In this paper, we use the proxy-protected signature scheme because the role of the proxy signers is not clear if we assume that the original signer can make proxy signature.

Bilinear Pairing: Consider a group $\mathbb{G}$ of problems for which it is hard to solve CDHP (Computational Diffie-Hellman Problem) but easy to solve DDHP (Decision Diffie-Hellman Problem). Those problems are called GDHP (Gap Diffie-Hellman Problem) [8], and the bilinear pairing is one of their solutions [9].

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be cyclic groups with order of a prime q. Let P be a generator of $\mathbb{G}_1$. If $\mathbb{G}_1$ is an additive group and $\mathbb{G}_2$ is a multiplicative group, the bilinear pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ that satisfies the following properties on elliptic curves over finite field.

- **Bilinear:** $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
- **Non-degenerate:** there exists $P, Q \in \mathbb{G}_1$ such that $\hat{e}(P, Q) \neq 1$.
- **Computable:** there exists an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

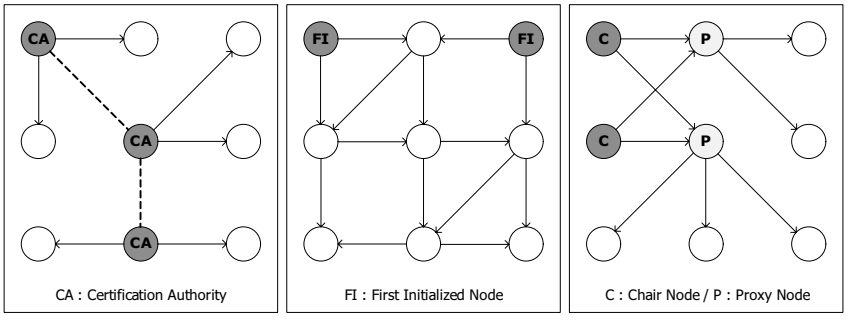
Using those properties, we can easily verify whether $xyP = zP$ or not when we have P, xP, yP and zP on the elliptic curves.

3.2 Basic Protocol

Our proposed scheme is based on the signature schemes and its relative methods described in [10] and [11]. We assume that a system authority exists at the beginning of the network operation. Also, all nodes in the network generate their public and private key pair in self-initialization. The following is the procedure of our scheme:

1) By using secret sharing scheme, the authority shares its partial private key with the first initialized nodes which are called chair nodes. Also, the authority directly assigns proxy signature keys to delegation nodes which are called proxy nodes.
2) By using threshold signature scheme, the chair nodes which are assigned partial private key from the system authority can assign proxy signature key to a proxy node without the system authority.
3) A proxy node which has proxy signature key can issue proxy certificates for normal nodes which should receive their proxy certificates from the proxy nodes only.

A simple illustration of our scheme versus other schemes (i.e. multiple CA and diffusion wave) is in Fig. 1(c).



(a) Multiple CA (b) Diffusion Wave (c) Our Scheme

Fig. 1. Certificate Issuing Schemes

Common Parameters: Consider PK_o and SK_o such that $PK_o = SK_oP \in \mathbb{G}_1$ are the system public and private key pair (P is a generator of group $\mathbb{G}_1$). Also, consider PK_p and SK_p such that $PK_p = SK_pP \in \mathbb{G}_1$ are proxy node's public and private key pair. Then, $(\mathbb{G}_1, \mathbb{G}_2, \hat{e}, q, P, \mathbf{H}_1, \mathbf{H}_2)$ are common parameters where $\mathbf{H}_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ and $\mathbf{H}_2 : \{0,1\}^* \rightarrow \mathbb{G}_2$ are two different hash functions.

Proxy Signature Key: The system authority generates a $(t-1)$ degree polynomial $f(x) = SK_o + a_1x + \cdots + a_{t-1}x^{t-1}$ with random coefficients $a_1, a_2, \cdots, a_{t-1} \in_R \mathbb{Z}_q^*$ and the private key SK_o. For all chair nodes $i \in I$ where $I = \{1, 2, \cdots, n\}$, the authority distributes $x_i = f(i)$. If a node collects all elements of a subset $S \subset I$ such that $|S| \geq t$, it can recover the secret key SK_o by using Lagrange interpolation in which $SK_o = \sum_{i \in S} L_i x_i$ where $L_i = \prod_{j \in S} -x_j/(x_i - x_j)$.

The proxy signature key can be assigned for the proxy node by two ways. In the first way, it is assigned by the system authority directly while in the other way it is assigned by the chair nodes.

- From **the system authority:** the system authority makes a warrant w and computes its hash value $\mathbf{H}_2(w)$. The authority then assigns w and $SK_{o_w} = SK_o\mathbf{H}_2(w)$ to a proxy node.
- From **the chair nodes:** the chair node i makes a warrant w for a proxy node and computes $x_i\mathbf{H}_2(w)$. The chair node i then sends the result to the proxy node. The proxy node can make the proxy signature key such that $SK_{o_w} = SK_o\mathbf{H}_2(w) = \sum_{i \in S} L_i x_i \mathbf{H}_2(w)$ if it gathers the partial signatures over t. In this case, the warrant w must not include specific information of the chair nodes for using threshold signature scheme.

The proxy node should check if $\hat{e}(SK_{o_w}, P) = \hat{e}(\mathbf{H}_2(w), PK_o)$. Only when this is satisfied, the proxy node makes proxy signature key $SK_w = SK_{o_w} + SK_p\mathbf{H}_2(w) = (SK_o + SK_p)\mathbf{H}_2(w)$. If DLP (Discrete Logarithm Problem) is hard to solve, then SK_o is hard to be revealed during those process.

Certificate Issuing Through the Proxy Node: A node's certificate that includes its self-initialized public key can be issued by a proxy node only. A proxy node generates a valid proxy certificate $\langle \Lambda, c, U, w \rangle$ for the unsigned certificate Λ where $c = \mathbf{H}_1(\Lambda \| \hat{e}(P,P)^r)$, $r \in_R \mathbb{Z}_q^*$, and $U = cSK_w + rP$. In addition, r, $\hat{e}(P,P)^r$, and rP can be computed in the idle time so that a proxy node needs only one hash computation, one addition, and one multiplication operations on the elliptic curves in real time for signing a certificate.

Certificate Verifying: A node needs to verify other nodes' certificates to communicate with them trustfully. To verify a certificate $\langle \Lambda, c, U, w \rangle$, a node can be sure of the validation of a certificate by checking if equality in Equation (1) is satisfied.

$$c = \mathbf{H}_1(\Lambda \| \hat{e}(U,P)\hat{e}(\mathbf{H}_2(w), PK_o + PK_p)^{-c}) \tag{1}$$

3.3 Other Certificate Management Issues

Certificate Revocation: If the chair or other proxy nodes find a misbehaving node that is harmful to the system or the other nodes, they may exchange that information through the CRL (Certificate Revocation List). There is the information about issuer (e.g. the proxy node) of a revoked certificate in the CRL so that other nodes can know what proxy node mass-produces the certificates for the misbehaving nodes. Therefore, the discipline measure for that proxy node may be desired.

Delegation Deprivation: If some chair nodes want to deprive the authority of a proxy node, they should firstly make a report signed by system private key to notify that report. To do so, they should cooperate with other chair nodes to gather partial reports. Through this cooperation, threshold signature can be used. The report to notify the deprivation has to flood into network for all nodes. If a node receives the report that has the same issuer information with its certificate's issuer, the node must revoke its certificate and try to obtain a new certificate from other proxy nodes.

Also, the chair nodes should assign new proxy nodes when the number of proxy nodes decreases below a specific number.

Certificate Renewal: If the valid date of a certificate is expired or the corresponding deprivation report is arrived, a node can obtain a new certificate from the proxy nodes. If necessary, a node can obtain several certificates in anticipation.

Selection of a New Chair Node: The number of the nodes that have the partial system private keys must be over the threshold value t. Otherwise, no more nodes can be proxy nodes. So that, the chair and proxy nodes cannot maintain their interoperability. Therefore, a new chair node need to be selected though extremely strict authentication rules and a new partial private key should be assigned to that chair node.

4 Simulation

4.1 Simulation Environment

In oder to analyze the performance of our scheme, we built our simulation on the OMNeT++. The simulation parameters are shown in Table 1.

Table 1. Simulation Parameters

Parameters		Explain
Playground Size		300×300m^2 600×600m^2
Data Link		IEEE 802.11
Max Interference		74.142m
Routing		AODV-UU
Transport		UDP
Mobility	Type	random way point
	Moving Speed	up to 10 m/s
Num. of Nodes	Diffusion Wave	50 (first initialized nodes: 15)
	Our Scheme	50 (chair nodes: 10, proxy nodes: 5)
Message Length		4096 bits
Schedule Frequency		1 time per 5 $\sim$ 10 sec
Simulation Time		up to 43,200 sec

Playground Size and Coverage: Let N_t be the total number of nodes, P_{size} be the size of playground, and C_{size} be the coverage area of a node. The average number of neighbor nodes N_n can be obtained as in Equation (2).

$$N_n = \frac{N_t}{P_{size}} \times C_{size} = \frac{N_t}{P_{size}} \times \pi r^2 \tag{2}$$

where r is the maximum interference distance of a node. Note that, N_t/P_{size} refers to the node density on the whole area of playground.

If the playground size P_{size} is 300×300m^2 and the maximum distance for transmission r is about 74m, a node can directly communicate with 9.55 nodes on average. In 600×600m^2 playground, however, a node can directly communicate with only about 2.38 nodes on average.

Mobility and Communication Chances: In our simulation, the node's moving speed increases from 1 m/s to 10 m/s. Even though we assume that one node has a fixed position and another node directly passes through the surrounding of that node, the maximum connection time is 148 seconds in 1 m/s and 14.8 seconds in 10 m/s.

$$\text{Mean Distance} = 2r \times \frac{2}{\pi} \int_0^{\frac{\pi}{2}} \sin\theta d\theta = \frac{4r}{\pi} \tag{3}$$

Therefore, the distance for passing the circumference of a node is about 94.2m on average. So that, the link between two nodes is kept for 94.2 seconds in 1 m/s and 9.4

seconds in 10 m/s. Since one message is sprang up every 7.5 seconds on average, a node has 12.5 chances of communication in 1 m/s and 1.3 chances in 10 m/s.

Transport Protocol: In our simulation, we use UDP (User Datagram Protocol) as a transport protocol instead of TCP (Transmission Control Protocol) even though our scheme needs high integrity. Our selection is based on that the congestion control mechanism of TCP unexpectedly makes lots of burden if the network channel is not error-free like the case of wireless communication environment[12,13]. The integrity of datagram can be guaranteed by re-transmission if the error is detected in the application layer. In spite of the fact that there are several suggestions for TCP implementation in wireless communication environment, we could not be sure of correctness of these suggestions and feasibility due to the un-integrity with our simulator.

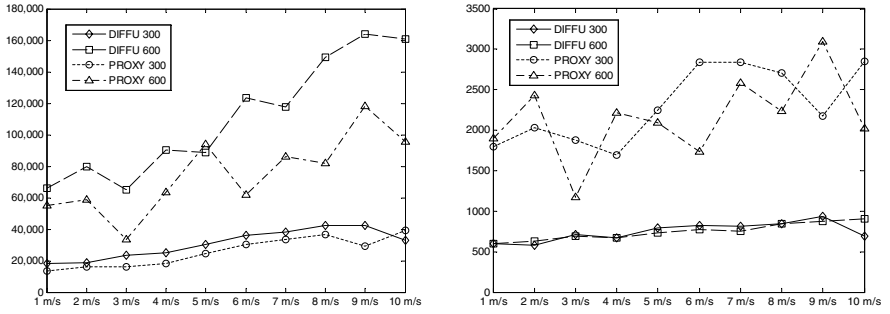
Message Frequency and Operation Time: Basically, the time for encryption and decryption on elliptic curves over finite filed is known to take hundreds of milliseconds on typical embedded devices including those with ARM-based micro controller [14]. In addition, we used the random delay up to 0.2 seconds for avoiding packet loose caused by radio interference in our simulation. Therefore, our assumption of $5\sim10$ seconds as the message schedule interval in the application layer makes the required time for encryption and decryption operations negligible. So that, those time parameters are expected not to make any congestion in the application layer.

4.2 Simulation Results: Network Performance

From the simulation results, as shown in Fig. 2(a), the total amount of generated packets in our scheme is much less than that of the diffusion wave. In both cases, however, the total amount of packets gradually increases by nodes' speed. On the contrary, we can see that the amount of unicast packets in our scheme is more than the diffusion wave in Fig. 2(b). The reason of more unicast packets is that our scheme in the simulation uses a number of unicast packets to inform proxy nodes' identifiers. Those informing packets seem to be significantly larger than partial signature gathering packets.

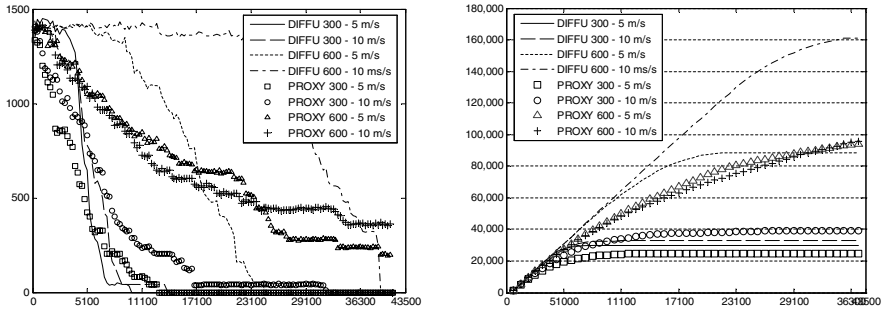
The number of stacked packets sent from UDP are shown in Fig. 3. In an early stage, the number of packets in our scheme is rapidly decreased because the proxy nodes can issue certificates for some nodes nearby them. On the contrary, the number of packets in the diffusion wave is kept in the similar level at the point of that time, but it is decreased at an accelerating pace after certain time because the nodes that can issue partial certificates for other nodes are increased. However, the diffusion wave does not work in $600\times600\text{m}^2$ playground environment by the moving speed of 10 m/s. The overall number of packets in our scheme decreases linearly by the lapse of time. The reason of why there is some long intervals in which the number of packets is kept in the same level is due to that a few number of nodes suffer from its bad environment caused by the random movement for searching the proxy nodes.

The increase of the number of nodes that have a complete certificate also reflects the difference between the diffusion wave and our scheme as shown in Fig. 4. The shape of Fig. 4 is very similar with Fig. 3(a) because the nodes that obtain a complete certificate



(a) total packets (b) unicast packets

Fig. 2. The total amount of packets sent from UDP according to the increment of moving speed



(a) 300 seconds stacked (b) totally stacked

Fig. 3. The number of stacked packets send from UDP by the lapse of time

do not make the broadcast messages any more. As shown in Fig. 2(b), the number of the unicast messages is very small as compared with the number of the broadcast messages. Therefore, the main reasons that the diffusion wave method suffers from lots of network traffic is not only due to transmitting several partial certificates, but also for finding the nodes that can issue partial certificates.

4.3 Simulation Results: Node Performance

The diffusion wave requires more decryption even though the proxy nodes should sign as much as the number of normal nodes in our scheme. In our simulation, one node in the diffusion wave signs about 3 times while one proxy node signs about 7 times on average. However, the overall number of times for signing certificates in the diffusion wave is about 5 times greater than in our scheme. If the threshold is lager than our simulation threshold parameter, the burden will be larger. This situation is shown in Fig. 5.

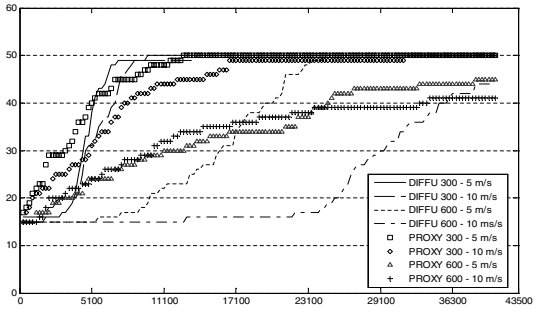


Fig. 4. The number of nodes which have a complete certificate by the lapse of time

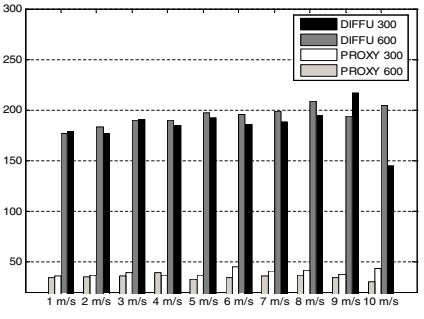


Fig. 5. The number of times for signing certificates in each scenario

5 Protocol Evaluation

5.1 Flexibility

Our proposed scheme is similar to multiple CA model (illustrated in Fig. 1(a)) in that both have several certificate issuers. However, the proxy nodes in our scheme are transfered certificate issuing competence from one authority (i.e. system authority) on contrary with multiple CA model which has several authorities. A node in multiple CA model should request its CA the validations of other nodes' certificates. So that, the CAs in multiple CA model should prepare the certificate chain to other CAs. However, in our scheme, there is not need for such request due to the warrant from system authority though a node also should know other proxy nodes' public keys. The validations of proxy nodes' public keys are guaranteed by the warrant signed by system private key.

5.2 Network Performance

From the shown simulation results, it is clear that the diffusion wave takes much time for obtaining first authentication (i.e. a complete certificate). However, the diffusion wave also has the positive effect such that it shortens the time to take a complete certificate with increasing speed. Instead of that effect, many nodes make lots of network traffic for

finding the nodes which are available in service and for requesting the several partial certificates. In addition, the diffusion wave provide stable and equal chances for all nodes to take their certificate in a good network environment. However, it is very hard to obtain a complete certificate in bad network environment.

The proxy signature method requires less traffic for issuing the certificates than the diffusion wave but there is a large difference of difficulty to obtain certificate for each node. Also, if the system authority at the beginning of the network operation initialize a few chair nodes only but any proxy nodes, the network will not work because of the rarity of the proxy nodes. Also, proxy nodes can exist after the chair nodes distribute the proxy signature keys for the proxy nodes. Therefore, even though we did not simulate that case, it is expected that the system authority must initialize not only the chair nodes but also the proxy nodes.

5.3 Security Consideration

There are many direct or indirect security issues about certificate management in ad hoc network, but we consider two direct security issues that the diffusion wave holds but our scheme mitigates.

Random Attack on the Network: Let τ be the threshold for recovering the secret, N_s be the number of nodes that have the secret, and N_a be the number of nodes that the attacker can attack. The attacker cannot obtain any information from neighbor nodes by attacking directly.

If the attacker cannot analyze the network traffic so that he cannot attack the nodes selectively(or simple, if the nodes behave very carefully for hiding their roles from the attackers), the attacker can try to attack only randomly selected nodes. Therefore, the expected number of nodes that have the secret is only $(N_s N_a)/N_t$ when N_a number of nodes are randomly selected from N_t nodes. Since the secret can be recovered only if $(N_s N_a)/N_t \geq \tau$, the attacker needs to attack N_a number of nodes shown in Equation (4) to obtain the system secret on average.

$$N_a \geq \frac{N_t \tau}{N_s} \tag{4}$$

Based on that, in the diffusion wave, the attacker can recover the system secret if the attacker successfully attacks only τ nodes because $N_s \simeq N_t$.

In our scheme, the probability p that an attacker can succeed in performing such an attack is shown in Equation (5) because all nodes selected by the attack should have the secret. Fig. 6 shows those probabilities.

$$\Pr[p] = \frac{N_s}{n} \frac{(N_s - 1)}{(N_t - 1)} \frac{(N_s - 2)}{(N_t - 2)} \cdots \frac{(N_s - \tau + 1)}{(N_t - \tau + 1)} = \frac{N_s!(N_t - \tau)!}{N_t!(N_s - \tau)!} = \frac{{}_{N_s}P_\tau}{{}_{N_t}P_\tau} \tag{5}$$

Originality of the Authority: The authentication and the certificate issuing are totally two different operations. However, we can consider that the certificate issuing is a part of the authentication if the authentication is a condition for obtaining the certificate.

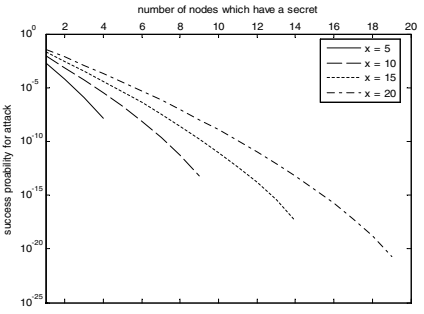
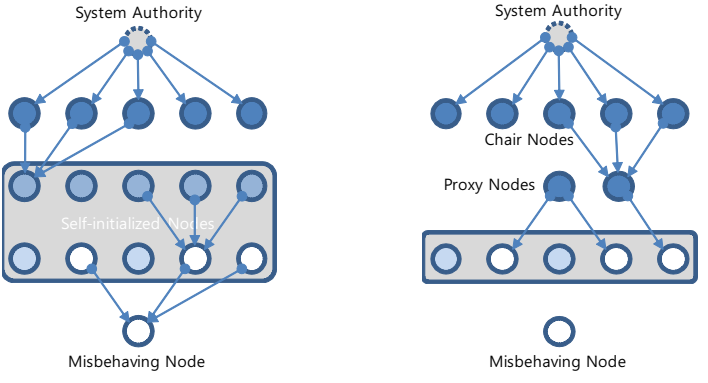


Fig. 6. The success probability for attack with the threshold and the number of nodes that have the secret



(a) diffusion wave (b) proxy signature

Fig. 7. The problem of the originality of the authority

Therefore, we can state that the proxy nodes have a part of the authority for authentication. In that point of view, the discussion about the originality of the authority is required.

Let us refer to nodes initialized by the system authority at the beginning of the network operation as the $1^{\mathbf{st}}$ generation and those authenticated by the nodes of $1^{\mathbf{st}}$ generation as the $2^{\mathbf{nd}}$ generation and so on. If the nodes in certain generation are authenticated by the nodes of previous generation, the originality of authority will degrade increasingly as the generation raises. That is, the misbehaving nodes can join the network legally because of low originality of authority even though the system authority initially does not want them to join in the network. The diffusion wave scheme (shown in Fig. 7(a)) does not consider this problem.

However, in our scheme (an illustration is shown in Fig. 7(b)), the nodes that can authenticate other nodes are only the proxy nodes authenticated by the system authority

or the chair nodes of 1^{st} generation. Therefore, the number of the generation does not exceed 3 in our scheme. So that, it is expected that the problem caused by the originality of the authority rarely happens.

6 Conclusion

In this paper, we introduced the certificate issuing scheme using proxy and threshold signatures for self-initialized ad hoc network. In our introduced scheme, chair nodes that can distribute partial proxy keys for proxy nodes are authenticated by the system authority. In addition, proxy nodes that can issue certificates for other nodes are authenticated and initialized by the system authority or the chair nodes. As it is demonstrated in the simulation results, our scheme has many advantages over the diffusion wave which is a main scheme for the certificate issuing in the self-initialized ad hoc network. If, however, the system authority does not authenticate and initialize any proxy nodes, it is expected that more time is required until there are enough number of the proxy nodes for well-operating network. Therefore, it is recommended that the system authority authenticate the proxy nodes with the chair nodes to save the time.

References

1. Capkun, S., Buttyán, L., Hubaux, J.-P.: Self-organized public-key management for mobile ad hoc networks. IEEE Transactions on Mobile Computing 2, 52–64 (2003)
2. Li, R., Li, J., Hisao, K., Liu, P.: Localized public-key management for mobile ad hoc networks. In: IEEE Global Telecommunications Conference, pp. 1639–1646 (2004)
3. Luo, H., Kong, J., Zerfos, P., Lu, S., Zhang, L.: Ursa: ubiquitous and robust access control for mobile ad hoc networks. IEEE/ACM Transactions on Networking 12, 1049–1063 (2004)
4. Li, J., Blake, C., Couto, D.S.J.D., Lee, H.I., Morris, R.: Capacity of ad hoc wireless networks. In: ACM Conference on Mobile Computing and Networking, pp. 61–69. ACM Press, New York (2001)
5. Shamir, A.: How to share a secret. Communications of ACM 22, 612–613 (1979)
6. Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 307–315. Springer, Heidelberg (1990)
7. Mambo, M., Usuda, K., Okamoto, E.: Proxy signatures for delegating signing operation. In: ACM Conference on Computer and Communications Security, pp. 48–57. ACM Press, New York (1996)
8. Okamoto, T., Pointcheval, D.: The gap-problems: A new class of problems for the security of cryptographic schemes. In: Kim, K.-c. (ed.) PKC 2001. LNCS, vol. 1992, pp. 104–118. Springer, Heidelberg (2001)
9. Menezes, A., Okamoto, T., Vanstone, S.A.: Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Transactions on Information Theory 39(5), 1639–1646 (1993)
10. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 31–46. Springer, Heidelberg (2002)
11. Zhang, F., Safavi-NainiTatsuaki, R., Lin, C.Y.: New proxy signature, proxy blind signature and proxy ring signature schemes from bilinear pairings. Cryptology ePrint Archive (2003)

12. Lakshman, T.V., Madhow, U.: The performance of tcp/ip for networks with high bandwidth-delay products and random loss. IEEE/ACM Trans. Netw. 5, 336–350 (1997)
13. Lefevre, F., Vivier, G.: Understanding tcps behavior over wireless links. In: IEEE Symposium on Communications and Vehicular Technology, SCVT-200, pp. 123–130. IEEE Computer Society Press, Los Alamitos (2000)
14. Gura, N., Patel, A., Wander, A., Eberle, H., Shantz, S.C.: Comparing elliptic curve cryptography and rsa on 8-bit cpus. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 119–132. Springer, Heidelberg (2004)

XWELL: A XML-Based Workflow Event Logging Mechanism and Language for Workflow Mining Systems

Min-Jae Park[1] and Kwang-Hoon Kim[2]

[1] Collaboration Technology Research Lab.
Department of Computer Science
Kyonggi University, South Korea
mjpark@kyonggi.ac.kr
[2] Collaboration Technology Research Lab.
Department of Computer Science
Kyonggi University, South Korea
kwang@kyonggi.ac.kr

Abstract. As the workflow/BPM systems and their applications are prevailing in the wide and variety industries, we can easily predict not only that very large-scale workflow systems (VLSW) become more prevalent and much more needed in the markets, but also that the quality of workflow (QOW) and its related topics be issued in the near future. Especially, in the QOW issues, such as workflow knowledges/intelligence, workflow validations, workflow verifications, workflow mining and workflow rediscovery problems, the toughest challenging and the most impeccable issue is the workflow mining and rediscovery problem, and its solutions should be based upon the event history information logged by workflow enactment engines. That is, the logged information is the most valuable one as well as the Alpha and the Omega of those QOW issues and solutions. In this paper[1], we propose a XML-based workflow event logging mechanism and language, which is called XWELL, and describe the implementation details of the mechanism so as to be embedded into the e-Chautauqua system that has been recently developed by the CTRL research group as a very large scale workflow management system. Finally, we explain how the XWELL will be applied and used to the workflow mining and rediscovery framework.

Keywords: Workflow Event Log Information, XML-based Log Schema, workflow Mining and Rediscovery, Quality of Workflow, EJB-based Very Large-Scale Workflow Management Systems.

1 Introduction

In order to improve the quality of workflows with a high, consistent, and predictable fashion, especially it is never enough to emphasize on the workflow traceability and rediscoverability: the workflows should be correctly designed; their execution should

[1] The research was supported by the special BEIT research fund of Kyonggi University. And the research result is based upon the BEIT's Grid Computing facility.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 900–909, 2007.
© Springer-Verlag Berlin Heidelberg 2007

be met their workload requirements; the workflows reasonably reflect their real-world business processes; and the (human or automated) process resources should be able to perform the work items in a timely fashion. But, many of the conventional workflow products have been experienced with being ineffective for the serious and large scale workflow applications, because they provide just the bare necessities of functionality - modeling capability, scripting tools, dynamic change supports, and many other basic features of the workflow systems. Particularly, many successful large scale workflow customers found out that as they tried to scale up from pilot test mode to enterprize-wide mode, there were severe symptoms of inflexibility, non-recoverability, and non-verifiability.[10][11][12] That is, it is necessary for the systems to be equipped with advanced features to effectively trace and observe the runtime behaviors of workflow procedures. The event logging mechanism ought to be the right solution for the advanced features like the workflows' runtime behaviors trace and observation, and the workflow quality improvement [12].

Based upon these practical backgrounds and motivations, this paper conceives a workflow execution event logging mechanism and implants it into the e-Chautauqua workflow management system [10] that has recently developed by the author's research group. Especially, in the mechanism, the workflow execution events log information [5][6] being stored by the system's engine components is formatted in XML. That is, we identified all types of events possibly happened in the system, classified them based upon the statuses of workflow instances running on and being managed by the system, and defined their XML schema to represent the log information.

Through the consecutive three sections, we introduce the overall description of the e-Chautauqua workflow management system, describe the details of the workflow event logging mechanism, and propose a XML-based event log language which is called XWELL. Finally, we summarize the implications of the mechanism by showing how the XML-based log language is used in the workflow mining and rediscovery framework.

2 The Overall System Architecture of e-Chautauqua

The author's research group has recently completed the development of a workflow management system that aims for very large scale workflows applications, and it is named e-Chautauqua workflow management system [10]. e-Chautauqua is based on the workcase-oriented workflow architecture [10], and especially we have implemented it by the Enterprize Java Beans framework approach, while almost all conventional workflow systems are based on the activity-oriented workflow architecture proposed by OMG [4][7][8][9]. This section shortly introduces e-Chautauqua's overall system architecture as an underlining workflow system of the XML-based workflow event logging mechanism to be described in the remaining sections.

As shown in Fig. 1, the functional descriptions of e-Chautauqua's architectural components and the ways they cooperate with each other are illustrated. Basically, e-Chautauqua is fully implemented so as to provide almost all services and facilities specified in the international workflow standards from the interfaces 1 through 5 announced by WfMC. For examples, in terms of the workflow definition language, it completely handles WPDL as well as XPDL, and also it is able to support the Wf-XML 2.0

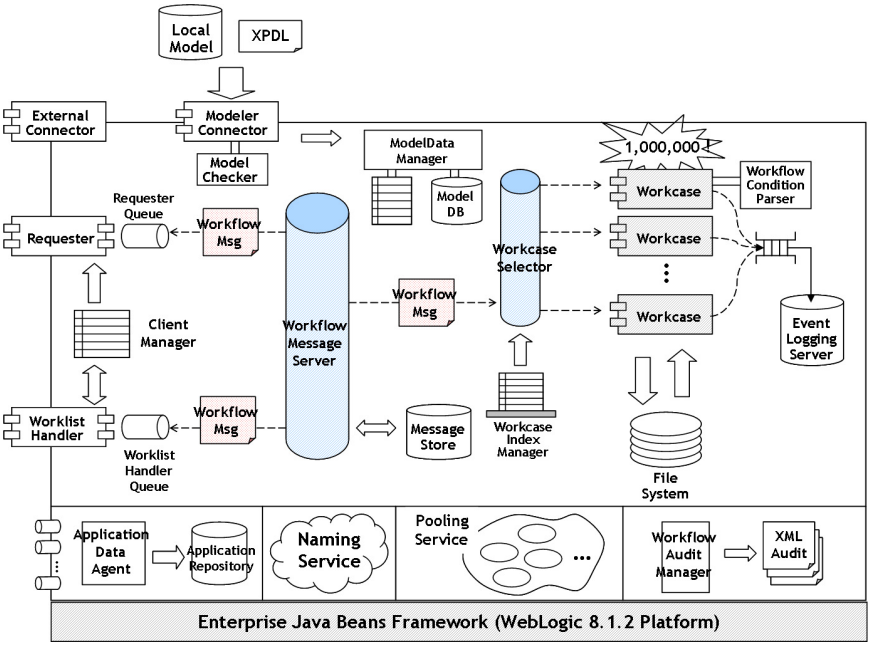


Fig. 1. The Overall System Architecture of e-Chautauqua

interoperability interface, which has recently released by WfMC, for interacting with other heterogeneous workflow systems. Our focus is on the relationship between the workflow enactment component (so-called workflow engine) and the log agent in charge of event logging functionality. The overall system architecture of e-Chautauqua, as depicted in Fig. 1, consists of several agent-based components, such as PDL (process definition language) Agent, OM (organization management) Agent, Administration Agent, Tool Agent, requester, worklist handler, script interpreter, Log Agent, workflow Message Beans using JMS message queue, and Enactment & Control Agent consisting of workcase objects registered in the workcase pool. These all components are deployed on an Enterprize Jave Beans middleware platform and they asynchronously communicate with each other through a JMS-based message queue mechanism named the workflow message beans.

3 The XML-Based Workflow Event Logging Mechanism and Language

As a functional part of the e-Chautauqua workflow management system, we have developed a XML-based event logging mechanism and language, which is called XWELL. In this section, we describe the functional structure of log agents, and explain about how the engine components take events, generate the events' log message formats and their language, and finally store them on log database. Finally, we also introduce the

asynchronous logging message queue mechanism that is used for the engine components to store their event log information formatted in XWELL.

3.1 Engine's Logging Components

The core engine components that distinguish e-Chautauqua's engine from other workflow engines are the workcase object components. That is, as stated in the previous section, almost all conventional workflow engines are based on the activity-oriented workflow architectural style [13], but on the other hand the e-Chautuaqua's engine is based on the workcase-oriented workflow architectural style [13]. In the activity-oriented workflow engines, the control flow management functionality is performed through interactions among activity objects, because the core component is the activity object component. In contrast to this, the control flow management is done by the workcase objects in the workcase-oriented workflow engine (e-Chautauqua), and so the activity precedence information is stored to the inside of each workcase object as data. Conclusively, the most valuable benefit that we expect from the workcase object component is that the control flow management functionality can be done by much simpler mechanism, and also it can be efficiently done with much smaller amount of computing resources, because the workcase-oriented workflow engine is able to dramatically decrease the number of objects resided and managed in the system. These facts mean also that the workcase-oriented workflow engine like e-Chautuaqua must be more appropriate and durable for the very large scale workflows, and it must give satisfaction in the higher degree of workflow instantiation complexity.

Fig. 2 is to present the relationship and interactions between the e-Chautauqua engine's workcase components and the log agent components that are in charge of the execution of the event logging mechanism proposed in this paper. As shown in the figure, once one of the workflow clients requests its assigned activity enactment services, then its corresponding workcase component performs the requested services, and makes their corresponding events log information according to the XML-based log message format to be specified in the next section and stores to log databases through the corre-

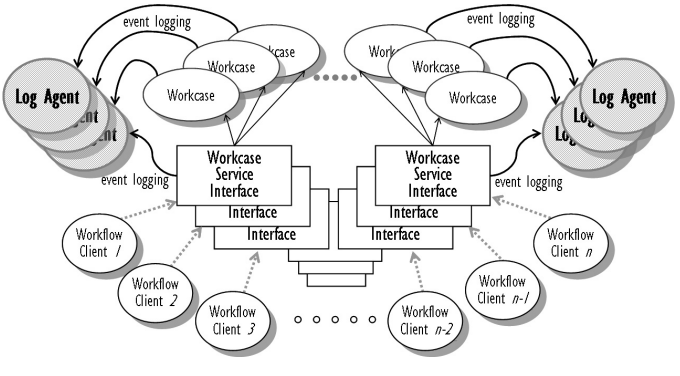


Fig. 2. e-Chautauqua Engine's Logging Components

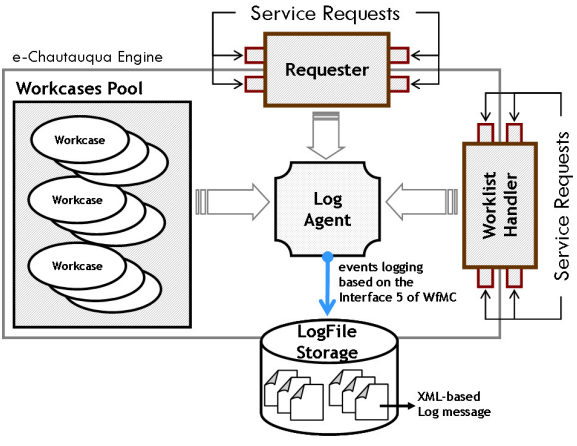


Fig. 3. e-Chautauqua's Events Logging Mechanism

sponding log agent. The detailed role and functionality of the log agent is described in the next section.

3.2 The Implementation Details of XWELL

All events coming from the e-Chautauqua workflow enactment components, such as worklist handler, requester and workcases, would have to be logged on a certain type of storage. These events log information is precisely well-defined in the audit and monitoring functions' standard specifications of WfMC [2][3]. However, these audit and monitoring information have not defined by any types of XML-based formats. So, in this section, we try to identify and classify all events produced by the engine components, and also to define them by a certain form of XML-based representation formats.

3.3 Structure of the Mechanism

The functional structure of the workflow event logging mechanism is depicted in Fig. 3. It consists of the following three types of components:

- *Event triggering components* — Requester and Worklist Handler
- *Event formatting components* — Workcase Pool
- *Event logging components* — Log Agent and Log File Storage

The *event triggering components* handle the workflow enactment services requested from workflow clients, and, these services are able to be categorized into three levels of classification — Workcase level class, Running activity level class, and Workitem level class. The *event formatting components* try to compose event log messages according to the service classes after performing the requested services. Finally, the *event logging*

Table 1. Workcase Level Class's Events Log Message and Its XML-based Language

Log Element	XML Tag	Description
WorkcaseLog	<WorkcaseLog> ... </WorkcaseLog>	Event Log on Workcase
WorkcaseID	<WorkcaseID> workcaseID </WorkcaseID>	Unique ID of current workcase
ParentWorkcaseID	<ParentWorkcaseID> ParentWorkcaseID </ParentWorkcaseID>	Unique ID of initial(parent) workcase
WorkcaseName	<WorkcaseName> WorkcaseName </WorkcaseName>	Name of current workcase
State	<State> State = { INACTIVE | ACTIVE | SUSPENDED | COMPLETED | TERMINATED | ABORTED } </State>	Current state of workcase
PackageID	<PackageID> PackageID </PackageID>	Package ID identifying the definition used for creating this workcase
WorkflowID	<WorkflowID> WorkflowID </WorkflowID>	Process Definition ID identifying the definition used for creating this workcase
EventCode	<EventCode> EventCode = { WMCreatedWorkcase | WMStartedWorkcase | WMChangedWorkcaseState | WMCompletedWorkcase | WMTerminatedWorkcase | WMAbortedWorkcase } </EventCode>	This message code is associated with event of workcase
EventTimestamp	<EventTimestamp> EventTimestamp </EventTimestamp>	Timestamp at the time when the event was recorded
CreatedTimestamp	<CreatedTimestamp> CreatedTimestamp </CreatedTimestamp>	Timestamp at the time when the workcase was created
StartTimestamp	<StartTimestamp> StartTimestamp </StartTimestamp>	Timestamp at the time when the workcase was started

components, especially the log agents, take in charge of the responsibility of the event logging mechanism. Once, a log agent receives event logs and then transforms them into XML-based log messages, and store the transformed messages onto the Log File Storage.

3.4 Workflow Events Classification and Language

As shortly explained in the previous section, the workcase components, which are taking a role of the event formatting component, compose event log messages after executing the requested services from the event triggering components — the requester and the worklist handler. After doing the formatting job, they transmit the formatted event log messages to the event logging components — the log agents. Based on the formatted messages, the log agents form the XML-based event log information. In order to efficiently perform these logging-related jobs, we classify the events into three levels of classes — workcase level event class, running activity level event class, and workitem level event class. These event classes' message formats and their XML-based languages are precisely described in Table 1, Table 2, and Table 3, respectively. As shown in the EventCode rows of the tables, the detailed names of the events that are captured and logged by the mechanism are summarized as the following:

– *Workcase Level Events*: WMCreatedWorkcase, WMStartedWorkcase, WMChangedWorkcaseState, WMCompletedWorkcase, WMTerminatedWorkcase, WMAbortedWorkcase

Table 2. Running Activity Level Class's Event Log Message and Its XML-based Language

Log Element	XML Tag	Description
RTActivityLog	<RTActivityLog> ... </RTActivityLog>	Event Log on Running Activity Instance
WorkcaseID	<WorkcaseID> WorkcaseID </WorkcaseID>	Unique ID of the workcase with which current activity instance is associated
State	<State> State = { INACTIVE | ACTIVE | SUSPENDED | COMPLETED | TERMINATED | ABORTED } </State>	Current state of current activity instance
PackageID	<PackageID> PackageID </PackageID>	Package ID identifying the definition used for creating this activity instance
WorkflowID	<WorkflowID> WorkflowID </WorkflowID>	Process Definition ID identifying the definition used for creating this activity instance
ActivityID	<ActivityID> ActivityID </ActivityID>	Activity Definition ID identifying the definition used for creating this activity instance
ActivityName	<ActivityName> ActivityName </ActivityName>	Activity Name identifying the definition used for creating this activity instance
ActivityType	<ActivityType> ActivityType </ActivityType>	Type for current activity instance
EventCode	<EventCode> EventCode = { WMChangedActivityInstanceState | WMCompletedActivityInstance | WMTerminatedActivityInstance | WMAbortedActivityInstance } </EventCode>	This message code is associated with event of this activity instance
EventTimestamp	<EventTimestamp> EventTimestamp </EventTimestamp>	Timestamp at the time when the event was recorded

Table 3. Workitem Level Class's Event Log Message and Its XML-based Language

Log Element	XML Tag	Description
WorkitemLog	<WorkitemLog> ... </WorkitemLog>	Event Log on Workitem
WorkitemID	<WorkitemID> WorkitemID </WorkitemID>	Unique ID for the workitem
PackageID	<PackageID> PackageID </PackageID>	Package ID identifying the definition used for creating this workitem
WorkflowID	<WorkflowID> WorkflowID </WorkflowID>	Process Definition ID identifying the definition used for creating this workitem
ActivityID	<ActivityID> ActivityID </ActivityID>	Activity Definition ID for base activity of this workitem
ActivityName	<ActivityName> ActivityName </ActivityName>	Activity Definition ID for base activity of this workitem
EventCode	<EventCode> EventCode = { WMAssignedWorkitem | WMGetWorkitem | WMChangedWorkitemState | WMCompletedWorkitem } </EventCode>	This message code is associated with event of this workitem
EventTimestamp	<EventTimestamp> EventTimestamp </EventTimestamp>	Timestamp at the time when the event was recorded
Performer	<Performer> Performer </Performer>	Performer ID of current workitem
State	<State> State = { INACTIVE | ACTIVE | SUSPENDED | COMPLETED | TERMINATED | ABORTED } </State>	Current state of current workitem

– *Running Activity Level Events*: WMChangedActivityInstanceState, WMCompletedActivityInstance, WMTerminatedActivityInstance, WMAbortedActivityInstance

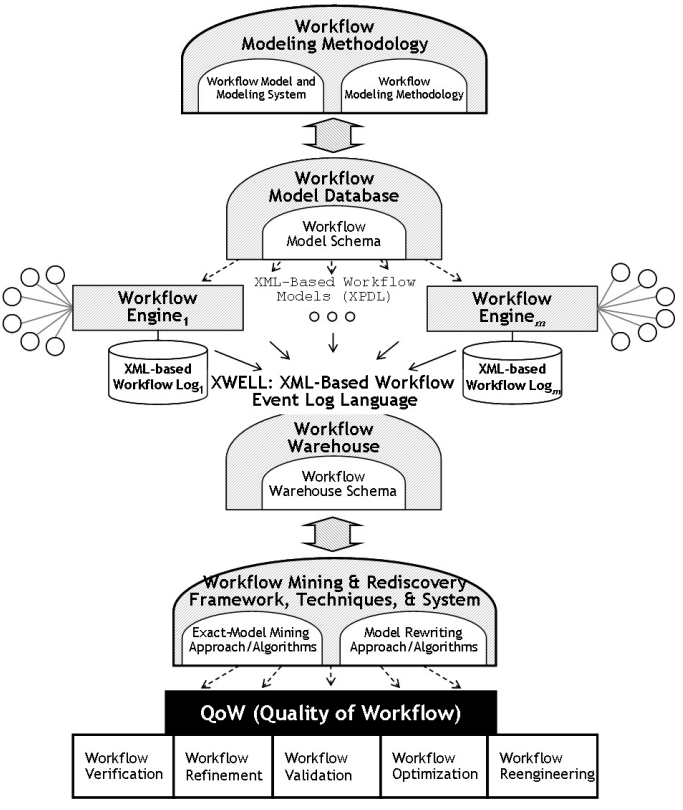


Fig. 4. A Workflow Quality Improvement Framework Through XWELL

– *Workitem Level Events*: WMAssignedWorkitem, WMGetWorkitem, WM ChangedWorkitemState, WMCompletedWorkitem

4 Implications of the Logging Mechanism on Workflow Mining and Rediscovery

The XML-based event log information generated by the logging mechanism is very important and useful knowledge, because the following crucial functionalities are possibly performed only through them:

– Workflow runtime status monitoring and reporting functionality
– Workflow runtime recovery functionality
– Workflow mining and rediscovery for quality improvement functionality

In this paper, however, our emphasis is placed on the quality of workflow as well as the workflow intelligence that aim at improving the quality of the corresponding workflow

model. Then, what is the quality of a workflow model? And how can we evaluate the quality of a workflow model? So, in this paper, we try to answer for those questions through the concept of XML-based event logging mechanism.

For that matter, this paper proposes a feasible framework coping with the workflow quality issue through the XML-based log information as shown in Fig. 6. The framework consists of modeling part and mining and rediscovery part for the quality of workflow. The workflow modeling part (the upper part of the figure) takes in charge of XML-based modeling and analyzing workflow models. The mining and rediscovery part (the lower part of the figure) is of rediscovering and/or mining the corresponding workflows from the workflow warehouse built from the XML-based workflow log information that is generated from the XML-based event logging mechanisms resided in the log agents of workflow engines. The workflow engines must be no matter whether they are homogeneous or heterogeneous if they embed a certain type of XML-based event logging mechanism handling with the event log message formats that are same with the XML-based event log message formats proposed in this paper.

5 Conclusions

In this paper, we have newly proposed the XML-based workflow logging mechanism and language. And, in order to show that the mechanism properly works in the real computing environment, we have implemented and implanted in the e-Chautauqua workflow management system that has recently implemented by the author's research group. Especially, the event level classes and their XML-based Language suggested in this paper should have to be a guideline for the XML-based workflow log message formats that are possibly referred by other workflow research groups who have plans to develop such a workflow logging mechanism. In recent, the literature needs various, advanced, and specialized workflow mining techniques and rediscovery algorithms that are used for finally providing feed-backs of their analysis results in order to the redesign and reengineering phase of the existing workflow and business process models. we strongly believe that this work might be one of those impeccable attempts and pioneering contributions for those efforts.

Acknowledgement. The research was supported by the special BEIT research fund of Kyonggi University. And the research result is based upon the BEIT's Grid Computing facility.

References

1. Alonso, D., et.al.: Failure Handling in Large Scale Workflow Management Systems. IBM Research Report RJ9913 (November 1994)
2. Alonso, D., et.al.: Exotica/FMQM: A Persistent Message Based Architecture for Distributed Workflow Management. In: Proceedings of the IFIPS Working Conference on Information Systems for Decentralized Organizations (August 1995)
3. Mani Chandy, K., Neuse, D.: Linearizer: A Heuristic Algorithm for Queuing Network Models of Computing Systems. Communication of the ACM 25(2) (1982)

4. Ellis, C.A., Morris, P.: The Information Control Nets Model. Performance Evaluation Review 8(3) (1979)
5. Ellis, C.: Team Automata. In: Proceedings of the ACM Group'97 Conference (November 1997)
6. Ellis, C., Maltzahn, C.: Chautauqua: A Flexible Workflow System. In: Proceedings of the 30th HICSS Conference (January 1997)
7. Ellis, C., Kim, K.: A Framework and Taxonomy for Workflow Architectures. In: The Fourth International Conference on Design for Cooperative Systems (May 2000)
8. Jablonski, S., Bussler, C.: MOBILE: A Modular Workflow Model and Architecture, University of Erlangen Internal Report (1995)
9. Kim, K.: Practical Experiences and Requirements on Workflow. In: Conen, W. (ed.) Coordination Technology for Collaborative Applications. LNCS, vol. 1364, pp. 145–160. Springer, Heidelberg (1998)
10. Kim, K., Ellis, C.: Performance Analytic Models and Analyses for Workflow Architectures. Information Systems Frontiers: A Journal 3(3), 339–355 (2001)
11. Miller, A., et.al.: CORBA Based Run Time Architectures for Workflow Management Systems, LSDIS Internal Lab. Report, University of Georgia (1996)
12. Perry, D., Wolf, A.: Foundations for the Study of Software Architectures. ACM SIGSOFT Software Engineering Notes 17(4) (1992)
13. Podgurski, A., Clarke, L.A.: A Formal Model of Program Dependencies and Its Implications for Software Testing, Debugging, and Maintenance. IEEE Trans. on Software Engineering 16(9) (1990)
14. Rolia, J.A.: Predicting the Performance of Software Systems. Technical Report CSRI-260, Computer Science Research Institute, University of Toronto (January 1992)
15. Rolia, J.A., Sevcik, K.C.: The Method of Layers. IEEE Transaction On Software Engineering 21(8) (1995)
16. Wallnau, K., Long, F., Earl, A.: Toward a Distributed Mediated Architecture for Enterprise-wide Workflow Management. In: Proceedings of the NSF Workshop on Workflow and Process Automation (May 1996)
17. Weissenfels, J., et al.: An Overview of the Mentor Architecture for Enterprise Wide Workflow Management. In: Proceedings of the NSF Workshop on Workflow and Process Automation (May 1996)
18. Ellis, C.A., Nutt, G.J.: Process Models and Systems for Logistics Support, A Proposal to the US DOD ARPA, CU Proposal No. 0595.00.0738B
19. Palaniswami, J., Lynch, I., Shevchenko, A.M., Reed-Forquet, L.: Web-based Multi-Paradigm Workflow Automation for Efficient Health-care Delivery. In: Proceedings of NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-Art and Future Directions (May 1996)
20. Chazin, D.: An Architecture and Transaction Model for Large Federated Databases. Ph. D. Thesis, University of Illinois at Urbana-Champaign (May 1996)
21. Alonso, G., Schek, H.-J.: Research Issues in Large Workflow Management Systems. In: Proceedings of NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-Art and Future Directions (May 1996)

Workcase-Oriented Workflow Enactment Components for Very Large Scale Workflows

Jae-Kang Won and Kwang-Hoon Kim

Collaboration Technology Research Lab.
Department of Computer Science
Kyonggi University, South Korea
jkwon@kyonggi.ac.kr, kwang@kyonggi.ac.kr

Abstract. The scalability problem in the current workflow management systems is the most impeccable issue, as the sizes of the systems and their applications are becoming much larger and more complex. Also, we can predict that very large-scale workflow systems will become more prevalent in the near future. Unfortunately, however, it is hard to say that the OMG's workflow enatment components[1] gives a satisfactory performance in coping with the issue. We so try to propose a new type of workflow enactment component—the workcase-oriented workflow enactment component—that is able to provide better performance in handling the scalability issue. In this paper[2], we describe the details of the workcase-oriented workflow enactment component and its related components—PDLAgent Component, OMAgent, WorklistHandler, Requester, Workflow Queue, Logger, ScriptInterpreter, and ToolAgent Component—that are additionally needed for performing the workflow enactment functionality implemented in an EJB-based workflow management system. Finally, we propose a possible overall system architecture to be implemented by applying the workcase-oriented workflow enactment components proposed in this paper.

Keywords: Workcase-oriented Workflow Enactment Component, Very Large Scale Workflow, Enterprize Java Beans Framework.

1 Introduction

In fact, workflow is becoming one of the hottest technologies in the enterprize information technology arena, which means that workflow systems have rapidly been adopted by many organizations. The primary reason why workflow systems are so hot-issued comes from the belief that they help large organizations to improve dramatically the way they operate, and their effectiveness has also come under our observation in numerous deployments. But, many of the workflow products have been experienced being ineffective for serious workflow applications because they generally lack necessities

[1] Jointflow: the activity-oriented workflow enactment metamodel and components as the de facto standard of Object Management Group.

[2] The research was supported by the special BEIT research fund of Kyonggi University. And the research result is based upon the BEIT's Grid Computing facility.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 910–919, 2007.
© Springer-Verlag Berlin Heidelberg 2007

for modeling capability, scripting tools, dynamic change supports, and many other advanced features of the workflow systems. Particularly, many successful large workflow customers found out that as they tried to scale up from pilot test mode to enterprize-wide mode, there were severe symptoms of inflexibility and non-scalability[10]. Therefore, the recent requirements on the workflow systems have been set up and characterized by the challenge - Performance and Scalability.

In terms of the more practical point of view, as organizations transfer more and more of their works onto their workflow systems, the workflow systems also ought to be much more scalable so as to accommodate much larger number of users, work items (workcases), information items, complex workflow applications, and so on. As a necessary consequence, the goal of this paper is to develop a new type of workflow enactment components to be composed of the workflow architecture and system that is satisfied with the scalability requirement not only covering from as small as 50 or 100 simultaneous workcases of a workflow procedure but also being scalable up to 500,000 or 1 million workcases without any further serious performance degradation. We name it "*workcase-oriented workflow architecture*." In this paper, we describe a set of core components to be used for implementing the new architecture through the component-based software design and development methodology.

Through the following section, we try to define the degree of workflow complexity for scoping our paper's goal. Next, we conceive a conceptual workflow architecture that is conceptually appropriate for very large scale workflow systems. And, we describe the implementation details of a set of components that is designed for implementing the new conceptual workflow architecture. Finally, we summarize the usability of the components by suggesting a possible system architecture applying the components, and conclude with its future works and further research issues.

2 The Scope of the Very Large Scale Workflows

Traditionally, the goal of a workflow management system was to orchestrate the enactment of workflow procedures ultimately being responsible for the control of either a single application domain's activities or a single organization's activities. However, recently it has been changing and scaling up in terms of the complexities in structure, size and workload, according for the workflow and business process technology to be flourishing in the almost all types of industries as well as in the almost all types of new process-driven business models, such as process portals, POD (Process-on-Demand), process choreography and orchestration, process collaboration, and inter-organizational workflows and business processes business models, which need to coordinate a huge number of instances spawned from variety sets of workflow procedures and business processes. That is, what we see happening now in the workflow literature is that the workflow technology and market might be in a stage of rapid scaling up, or at the point of transition from the end of the pilot test stage to the beginning of the cold reality stage. In other words, *the world of the Very Large Scale Workflows and Business Processes* has been already setting up in the market.

As a consequence of this atmosphere, we need to be concerned about the very large scale workflow and business process management systems. At the same time, it is

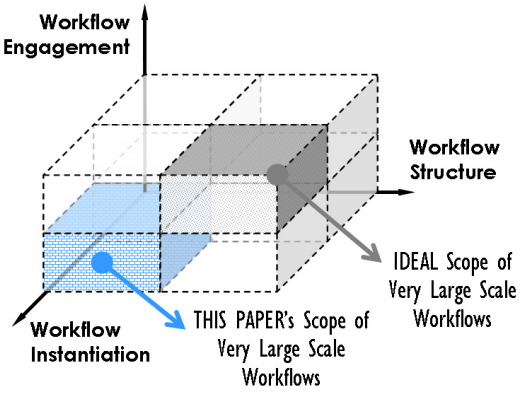


Fig. 1. The Degree of Workflow Scalability

necessary to make some criteria, which is called the degree of workflow Scalability, for effectively characterizing them. So, the very large scale workflows and business processes can be classified based on the degree of workflow Scalability consisting of three perspectives and dimensions as shown in Fig. 1 - *Workflow Engagement*, *Workflow Structure* and *Workflow Instantiation*. We describe the meanings of them as the followings:

- *Workflow Engagement*: This dimension of the workflow Scalability has to do with the sizes of workflow's physical components that are served and engaged for the workflow and business process management system. The physical components are directly related with the components of workflow models, such as actors, roles, activities, workflow procedures, business processes, and organizations. So, the degree of workflow commitment can be decided by the sizes of these components, and it also becomes an essential criterion that is able to characterize the system's scalability and performance - How well the workflow and business process management system handles, commits, and serves for a huge number of requests coming out of a large number of actors, procedures and organizational compartments, which are involved in one or even more organizations.
- *Workflow Structure*: This dimension of the workflow Scalability has something to do with interactional complexities between activities within a workflow procedure and between collaborative activities that are involved in interactions between workflow procedures in collaboration. In a large scale workflow procedure, several hundreds of activities might be co-associated with each other in diverse types of complex relationships, such as subworkflow, conjunctive, disjunctive, massively parallel, and cooperative relationships. Also, the recent e-Commerce, e-Logistics and e-Government marketplaces require more complex relationships for supporting the inter-organizational workflow services like collaborative workflows, choreographical workflows, orchestrated workflows, and chained workflows. So, these structural complexity ought to be a critical criterion for deciding the degree of workflow Scalability.

– *Workflow Instantiation*: This dimension of the workflow Scalability has directly related with measurement of the system's performance. In a large scale workflow application domain, the number of workcases instantiated from a workflow procedure may be from several hundred-thousands of workcases to several millions of workcases. Also, the large number of workflow procedures can be managed and maintained in a workflow management system. Specially, this dimension is inflicting heavy effect on the fundamental infrastructures of the workflow management systems. That is, the systems can be extensively deployed on distributed computing environments, such as clustered, grid, and peer-to-peer computing architectures, in order to handle those large scale workcases without any further performance degradation. So, this dimension should be the most important criterion in deciding the degree of workflow Scalability.

3 A Workcase-Oriented Workflow Architecture

Generally speaking, every information system is embedding a certain type of software architecture on behalf of the interactions and relationships among major components of the systems. Also, the architectural aspect of a system becomes one of the most important considerations that should be taken care for the efficiency and performance of the system. This section proposes a conceptual workflow architecture coping properly with the high-level degree of the workflow instantiation and scalability for the very large scale workflows, and describes the rationales for the architecture by considering the taxonomy[6] and the architectural performance estimation results[9] as evidences. Because of the page limitation, we would just specify the conceived result derived from the guidance[9].

In Fig. 2, we present a possible conceptual architecture preserving the workcase-oriented workflow architectural style. There need to be several different kinds of components in order to implement the architecture;The most essential component is ***workcase*** taking in charge of performing the control flow functionality that is the primary role of a workflow engine. So, the core work in implementing the architecture is on how to configure and organize the inside of the workcase component.

4 Components of the Workcase-Oriented Workflow Architecture

We have been working on developing a workflow management system that is coping with the scalability issue with respect to the very large scale workflows. The system supposes to be based on the workcase-oriented workflow architecture and also eventually to be implemented by the Enterprize Java Beans framework approach. In this section, we describe the details of a set of component types that is a sine qua non for successful implementation of the workcase-oriented workflow enactment architecture.

4.1 Major Component Types and Their Relationships

Based on the component-based software development methodology using the spiral model[3], we have developed a set of component types reflecting the workcase-oriented

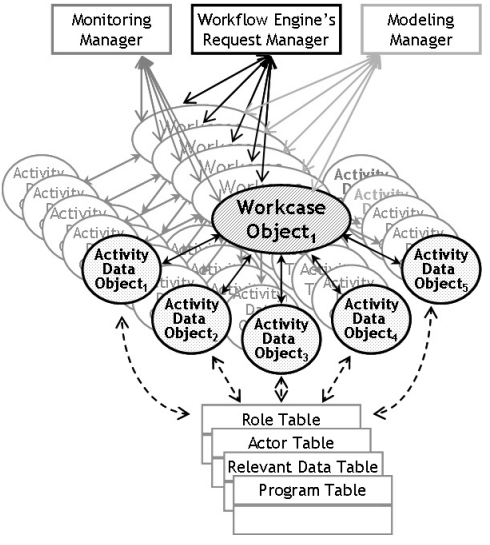


Fig. 2. The Workcase-oriented Workflow Architecture

Table 1. Types of Components and Their Layouts on EJB

Component Type	Component Layout	Description
PDLAgent	Stateless Session Bean	Handling Import/Export workflow models in XML-based Process Definition Language (XPDL)
OMAgent	Stateless Session Bean	Manipulating Organizational Information such as departments, roles, and actors
Requester	Stateless Session Bean	Handling WAPIs such as Connection, Process Control, Activity Control, Process Status, and Activity Status
Worklist Handler	Stateless Session Bean	Handling WAPIs manipulating the runtime clients' worklists
Workcase	Stateful Session Bean	Executing objects of Workflow instances
Workflow Queue	Message-Driven Bean	Message queue objects
Script Interpreter	Class Module	Handling conditional scripts on control flow transitions of OR-split
Tool Agent	Stateless Session Bean	Handling WAPIs bridging the engine/the runtime clients and invoked applications
Logger	Using Open Source Library (Log4j)	Handling WAPIs logging the engine's runtime events

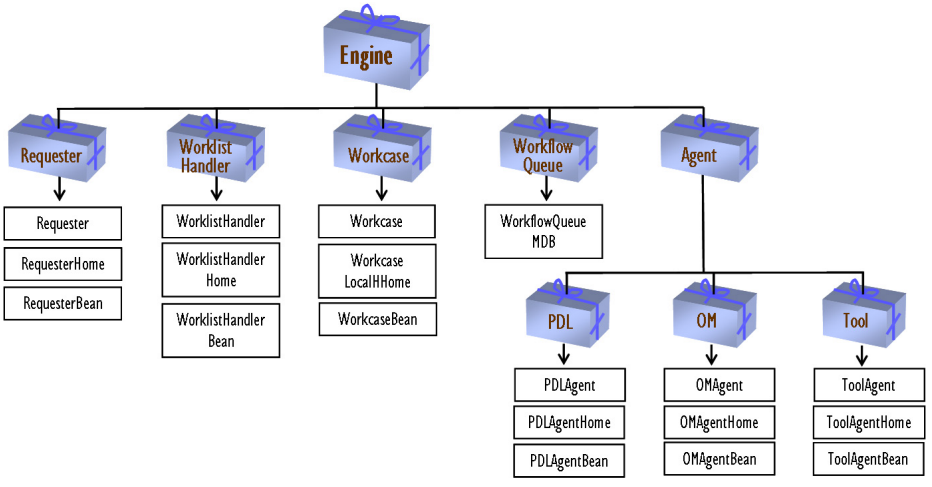


Fig. 3. The Package Structure of the Engine-related Components

workflow architecture. The core of the component types ought to be the workcase component as mentioned in the previous section, and its related component types are PDLAgent Component, OMAgent, Worklist Handler, Requester, Workflow Queue, Logger, Script Interpreter, and ToolAgent Component, which are additionally needed for performing the workflow enactment functionality. Table 1 is to represent the component types and their layout properties to be deployed on an enterprize java beans framework.

These engine-related component types are packed into the engine package as shown in Fig. 3. As shown in the figure, each of the component types being characterized by the session bean layout property consists of three types of classes—xxx, xxxHome, and xxxBean, where xxx is the name of a component type; The detail contents of the engine package structure are as the following:

- Requester Component Type: Requester, RequesterHome, RequesterBean
- Worklist Handler Component Type: WorklistHandler, WorklistHandlerHome, WorklistHandlerBean
- Workcase Component Type: Workcase, WorkcaseLocalHome, WorkcaseBean
- Workflow Queue Component Type: WorkflowQueueMDB
- Agent
  - PDLAgent Component Type: PDLAgent, PDLAgentHome, PDLAgentBean
  - OMAgent Component Type: OMAgent, OMAgentHome, OMAgentBean
  - ToolAgent Component Type: ToolAgent, ToolAgentHome, ToolAgentBean

Also, we have developed a set of data-related component types—Workflow Model, Organizational Model, and Run-time Client Component Types—from the workflow metamodel[8,9] as shown in Fig. 4. the detail contents of the data package structure are as the following:

- Workflow Model Data Component Type: ModelPackage, ModelWorkflowParameter, ModelAppParameter, ModelRelevantData, ModelActivity, ModelActivityTool, Model-

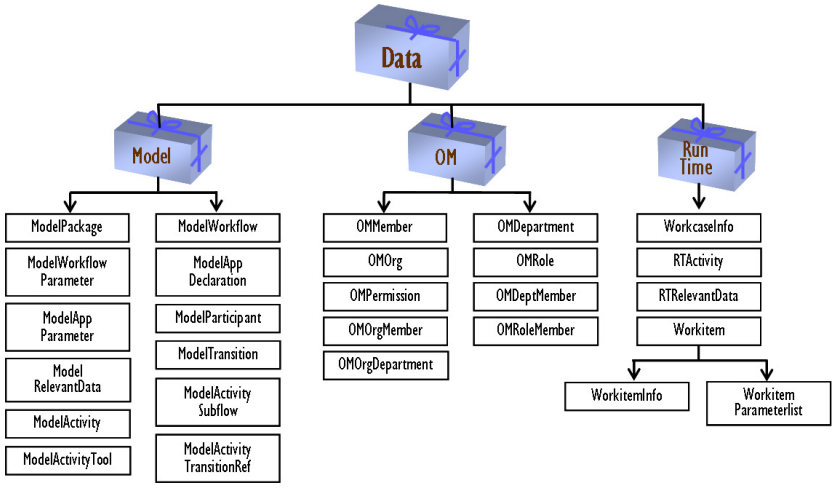


Fig. 4. The Package Structure of the Data-related Components

Workflow, ModelAppDeclaration, ModelParticipant, ModelTransition, ModelActivity-Subflow, ModelActivityTransitionRef

- Organizational Model Data Component Type: OMMember, OMOrg, OMPermission, OMOrgMember, OMOrgDepartment, OMDepartment, OMRole, OMDeptMember, OMDeptMember, OMRoleMember
- Run-time Client Data Component Type: WorkcaseInfo, RTActivity, RTRelevantData, Workitem (WorkitemInfo, WorkitemParameterlist)

The core one that distinguishes the engine package from other workflow engines' is the workcase component type. That is, as stated in the conceptual architecture section, almost all conventional workflow engines are based on the activity-oriented workflow architectural style[6], but, on the other hand, the engine package is based on the workcase-oriented workflow architectural style. In the activity-oriented workflow enactment engines, the control flow management functionality is performed through interactions among activity objects, because the core component is the activity component type. In contrast to this, the control flow management is done by the workcase objects in the workcase-oriented workflow enactment engine, and so the activity precedence information is stored on the inside of each workcase object as data. Conclusively, the most valuable benefit that we expect from the workcase component type is that the control flow management functionality can be done by the much simpler mechanism, and also it can be efficiently done with the much smaller amount of computing resources, because the workcase-oriented workflow enactment engine is able to dramatically decrease the number of objects resided and managed in the system. These facts mean also that the workcase-oriented workflow enactment engine must be more appropriate and durable for the very large scale workflows, and we expect that it gives a satisfaction in the higher degree of workflow scalability.

Table 2. Methods of the Core Engine Components

Return Value	Method Name	Input Parameters
in Requester		
Session	connect	String user-id, String passwd;
boolean	disConnect	Session session;
ModelWorkflow[]	getWorkflowList	Session session;
ModelWorkflow	getWorkflow	Session session, String workflow-id;
ModelWorkflowState[]	getWorkflowStateList	Session session;
ModelWorkflowState	getWorkflowState	Session session, String workflow-id;
boolean	changeWorkflowState	Session session, String workflow-id, String state;
String (workcase-id)	createWorkcase	Session session, String workflow-id, String workcase-name;
boolean	startWorkcase	Session session, String workcase-id;
RTWorkcase[]	getWorkcaseList	Session session;
RTWorkcase	getWorkcase	Session session, String workcase-id;
boolean	terminateWorkcase	Session session, String workcase-id;
RTWorkcaseState[]	getWorkcaseStatesList	Session session;
RTWorkcaseState	getWorkcaseState	Session session;
boolean	changeWorkcaseState	Session session, String workcase-id, String state;
RTActivity[]	getActivityInstancesList	Session session, String workcase-id;
RTActivity	getActivityInstance	Session session, String workcase-id, String activity-inst-id;
RTActivityState[]	getActivityInstancesStateList	Session session, String workcase-id;
RTActivityState	getActivityInstanceState	Session session, String workcase-id, String activity-inst-id;
boolean	changeActivityInstanceState	Session session, String workcase-id, String activity-inst-id, String state;
in Worklist Handler		
Workitem[]	getWorklist	Session session;
Workitem	getWorkitem	Session session, String workitem-id;
boolean	completeWorkitem	Session session, Workitem workitem;
in Workcase		
boolean	start	String workcase-id;
boolean	completeActivity	Workitem workitem;
void	runActivity	String workcase-id, ModelActivity exec-activity;

We would briefly describe the functionalities of other engine-related component types by introducing their functional methods and input/output parameters. In Table 2, we specify the functional methods of three major engine-related component types—Requester, Worklist Handler, and Workcase component types. As you see through the table, the requester plays a very important role as a bridge that deliveries several different types of requests between the inside components—Workcase, Workflow Queue, and Worklist Handler—and the outside components—PDLAgent, OMAgent, ToolAgent, and Run-time Client—of the engine part.

Finally, these all components communicate with each other through the workflow queue component type. The workflow queue components are realized as message-driven beans providing the reliable message transmission, which will be implemented by using

the JMS (Java Message Server) queue with an asynchronous message communication channel. For example, a component sending a message in a certain format transmits to a JMS queue, and then the JMS queue looks up the message map to find out the target component and its method and parameters by interpreting the received message from the sending component, and finally the JMS queue invokes the corresponding method either for passing the parameters' values or for returning the execution results.

5 Conclusions

In this paper, we have newly proposed the conceptual workflow architecture that is able to possibly realizing the high level of workflow scalability with respect to reasonably processing the very large scale workflows, which is called workcase-oriented workflow architecture. The architecture is based upon the proved belief that the instance-active workflow architectural style is much more scalable than any other styles of workflow architectures such as the server-client workflow architectural style and the class active workflow architectural style. By considering the properties of the workcase-oriented workflow architecture, we have derived a set of component types by applying the component-based software development methodology. The component types are sine qua nons for successful implementation of the architecture. We also have introduced the detail specifications, descriptions, and implementations of the component types. Based upon the results, we have designed a possible overall system architecture configuring the component types and their objects on an enterprize java beans framework. Finally, we will dig further our research into the world of very large scale workflows for not only exploring more advanced very large scale workflow architectures but also implementing very large scale workflow systems.

Acknowledgement. The research was supported by the special BEIT research fund of Kyonggi University. And the research result is based upon the BEIT's Grid Computing facility.

References

1. Alonso, D., et.al.: Exotica/FMQM: A Persistent Message Based Architecture for Distributed Workflow Management. In: Proceedings of the IFIPS Working Conference on Information Systems for Decentralized Organizations (1995)
2. Alonso, G., Schek, H.-J.: Research Issues in Large Workflow Management Systems. In: Proceedings of NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-Art and Future Directions (1996)
3. Boehm, B.W.: A Spiral Model of Software Development and Enhancement. IEEE Computer May 1998, 61–72 (1988)
4. Ellis, C.: Team Automata. In: Proceedings of the ACM Group'97 Conference (1997)
5. Ellis, C., Maltzahn, C.: Chautauqua: A Flexible Workflow System. In: Proceedings of the 30th HICSS Conference (1997)
6. Ellis, C., Kim, K.: A Framework and Taxonomy for Workflow Architectures. In: The Proceedings of ACM GROUP2000: the Fouth International Conference on Design for Cooperative Systems (2000)

7. Jablonski, S., Bussler, C.: 'MOBILE: A Modular Workflow Model and Architecture, University of Erlangen Internal Report (1995)
8. Kim, K.: Practical Experiences and Requirements on Workflow. In: Conen, W. (ed.) Coordination Technology for Collaborative Applications. LNCS, vol. 1364, pp. 145–160. Springer, Heidelberg (1998)
9. Kim, K., Ellis, C.: Performance Analytic Models and Analyses for Workflow Architectures. Information Systems Frontiers: A Journal 3(3), 339–355 (2001)
10. Kim, K.-H., Ahn, H.-J.: An EJB-based Very Large Scale Workflow System and Its Performance Measurement. In: Fan, W., Wu, Z., Yang, J. (eds.) WAIM 2005. LNCS, vol. 3739, pp. 526–537. Springer, Heidelberg (2005)
11. Miller, A., et.al.: CORBA Based Run Time Architectures for Workflow Management Systems, LSDIS Internal Lab. Report, University of Georgia (1996)
12. Perry, D., Wolf, A.: Foundations for the Study of Software Architectures. ACM SIGSOFT Software Engineering Notes 17(4) (1992)
13. Wallnau, K., Long, F., Earl, A.: Toward a Distributed Mediated Architecture for Enterprise-wide Workflow Management. In: Proceedings of the NSF Workshop on Workflow and Process Automation (1996)
14. Weissenfels, J., et al.: An Overview of the Mentor Architecture for Enterprise Wide Workflow Management. In: Proceedings of the NSF Workshop on Workflow and Process Automation (1996)

A Workcase-Based Distributed Workflow Architecture and Its Implementation Using Enterprize Java Beans Framework

Hyung-Jin Ahn and Kwang-Hoon Kim

Collaboration Technology Research Lab.
Department of Computer Science
Kyonggi University, South Korea
hjAhn@kyonggi.ac.kr, kwang@kyonggi.ac.kr

Abstract. This paper[1] gives the functional descriptions and implementation details of an distributed workflow management system that is especially aiming at the scalability issue with a feasible and reasonable performance for the very large scale workflow application domains. In order to cope with the scalability issue, we conceive a new conceptual workflow enactment architecture and implement the architecture through the enterprize java beans framework that provides a robust and efficient computing infrastructure for massive executable objects. Now, the system is completely operable on a distributed computing environment and is experimentally demonstrated with respect to giving the acceptable performance for the very large number of workcases (instances of workflow procedures).

Keywords: Degree of Workflow Scalability, Workcase-Oriented Conceptual Workflow Enactment Architecture, Enterprize Java Beans Framework, Very Large-Scale Distributed Workflow Management Systems.

1 Introduction

In the very large scale workflow domain, the degree of complexity that affects the system's scalability and performance can be determined by the factors—workflow engagement factor, workflow structure factor and workflow instantiation factor. The workflow engagement factor has to do with how large the organization is—How many workflow models including actors, roles and application programs does the system have to handle? The workflow structure factor has to do with how large and complex a single workflow model is—How many activities are associated within a single workflow model or collaborative workflows? and how complex their control/data flow relationships are? And, the workflow instantiation factor has to do with how large the number of instances coming out of workflow models is—How many instances do the workflow models produce? We, in this paper, try to estimate and measure the facts how well our architecture and system is coping with these factors, architecturally as well as systematically.

[1] The research was supported by the special BEIT research fund of Kyonggi University. And the research result is based upon the BEIT's Grid Computing facility.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 920–928, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In terms of the more practical point of view, as organizations transfer more and more of their paperwork to electronic systems, the workflow system must grow to accommodate much larger numbers of users, work items (workcases), information items, and complex workflow applications, in general. So, the goal is to develop a distributed workflow architecture and system that not only be useful for workflow applications as small as 50 or 100 simultaneous workcases of a workflow procedure, but also be scalable up to 500,000 or 1 million workcases without any seriously performable degradation. As a consequence of this, the architecture and system is able to provide acceptable performance for much larger and much more complex workflow applications, too. We name it 'e-Chautauqua[2] Workflow Management System'. The system is developed by the EJB-based framework approach, and it is fully deployable and operable now on the internet-based computing environment.

2 Workcase-Based Distributed Workflow Architecture

In this section, we analyze a set of distribution factors which may affect conceptual formation of workflow architecture, and conceive a conceptual workflow architecture by choosing a proper distribution factor—workcase—out of the analyzed factors.

2.1 Distribution Factors on Workflow Architectures

The distribution factors for workflow systems [7] consists of three levels of considerations—generic level, conceptual level, and implementation level. The general design issues are addressed at the top level (generic level), the Workflow specific design issues are considered at the second level (conceptual level), and the Specific implementation details such as hardware and network configurations are addressed at the bottom level (implementation level). Especially, the domain specific considerations are completely related with the conceptual level of the architectural framework. At a more detailed level, every workflow system deals with a set of conceptual objects such as workcases, activities, actors, and roles, and these objects and their representations are considered in the conceptual level of the architectural description.

The conceptual level of the framework elucidates the workflow architectural categories. So, the architectural properties for workflow systems shown Fig. 1 is based on three dimensions of the conceptual level considerations - process property, process creation and process control - for characterizing the behavior of a workflow engine. Through the taxonomy, we can construct many categories of workflow architectures, into which almost all workflow systems can be classified, and also it is reasonably imagine how we are feasibly shaping a conceptual architecture for very large scale workflow systems.

Using the process property dimension, we can characterize the properties of a workflow engine. That is, a workflow engine may perform its functionality by a monolithic process, and otherwise it may be composed of a set of processes (objects) that is based

[2] The origin in the Webster's dictionary: any of various traveling shows and local assemblies that flourished in the United States in the late 19th and early 20th centuries, that provided popular education combined with entertainment in the form of lectures, concerts, and plays, and that were modeled after activities at the Chautauqua Institution of western New York.

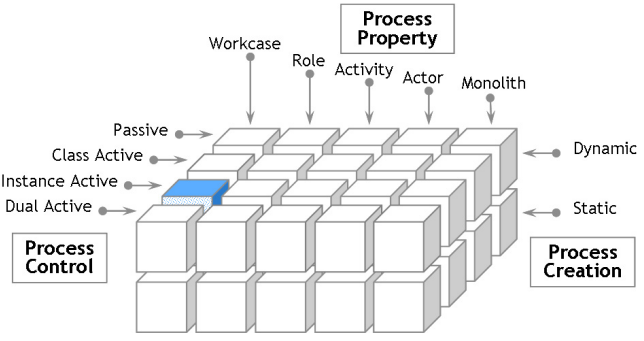


Fig. 1. The Distribution Factors of Workflow Architectures

upon one of the workflow entities - activity, role, actor, and workcase. These entities are so important because they are the domain specific properties that differentiate the conceptual architectures of workflow systems from other information processing systems' conceptual architectures. In the process control dimension, we can choose to implement control of workflow passively as data available on the server or elsewhere, or we can implement control actively with one active agent for each instance, or with one active agent for each class, or with both (dual active). Also, in terms of the process creation dimension, the processes that are created, manipulated, and destroyed in the active schemes can be static (created before execution begins), or dynamic (created during the execution of a workcase). Note that replicated or partitioned or hybrid schemes can be used, and the same dimensions can be used for data and scripts.

As an example that corresponds to a color-marked cubic in Fig. 1, let's consider selection of the instance active process control, workcase-oriented process property, and the dynamic process creation from our properties' choices. This selection implies an architecture in which the workcase is not simply represented as data, but as an executable process (object) that actively guides the workcase from activity to activity helping to find and select actors, roles, data, etc, for each activity. It can also act as an active negotiator for the workcase with the goal of expeditiously completing the task. It can thus help to perform activities, and perform negotiation on behalf of the workcase. We believe intuitively that the instance active model embodiment architectures are very promising for satisfying many advanced architectural requirements such as scalability, availability, and extensibility for very large scale workflow management systems.

2.2 The Conceptual Architecture

In Fig. 2, we present a possible conceptual architecture preserving the workcase-based workflow architectural style. There need to be several different kinds of agents in the architecture, and however the most essential agent is the control agent taking in charge of the control flow functionality that is the primary role of a workflow engine. So, the core work in deriving a workflow architecture is on how to configure and organize the inside of the control agent. Conclusively, in the workcase-oriented workflow architecture, a set of workcase objects fulfill the control agent's functionality, and a workcase object

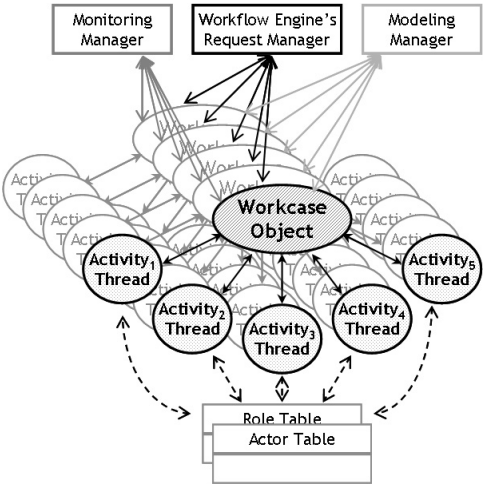


Fig. 2. The Conceptual Architecture of the Workcase-based Distributed Workflow Architecture

can be realized as either a work item (token) or an instance of workflow procedure in a workflow system. Therefore, it is possible to consider two kinds of workcase objects—One is the case that a token becomes an active component and takes in charge of activity enactment control by itself, and the other is the case that an instance of workflow procedure becomes an active component whenever a workcase is initiated. After completing an individual workcase, the active components (workcase objects) corresponding to the workcase are automatically destroyed. We believe that the workcase-oriented instance-active architecture should be very appropriate for maximizing the distribution capability and the resources' usage efficiency in performing the enactment scheduling and event-handling tasks. Note that almost all of the current workflow systems that are developed based on the OMG Joint-flow's workflow architectural style are categorized in the activity-oriented instance active workflow architectural style, and so their control agents are composed of a set of activity objects.

3 Implementation Details

We have developed a workflow management system that aims for very large scale workflows applications, and it is named e-Chautauqua workflow management system. e-Chautauqua is based on the workcase-oriented workflow architecture, and especially we have implemented it by the Enterprize Java Beans framework approach. In this section, we describe e-Chautauqua's overall system architecture and its workflow message queue mechanism providing asynchronous communication channels among the agents. Also, in order to show that e-Chatauqua might be reasonably acceptable as a very large scale worfklow management system, we gives two experimental observations on e-Chautauqua - One is an operational example by capturing the runtime client's screens, and the other is a performance evaluation result by measuring the response times for each of workcase objects to be created by the engine.

3.1 Overall System Architecture of e-Chautauqua

The functional descriptions of e-Chautauqua's architectural components and the ways they cooperate with each other are briefly explained in this section. Basically, e-Chautauqua is fully implemented so as to provide almost all services and facilities specified in the international workflow standards from the interfaces 1 through 5 announced by WfMC. For examples, in terms of the workflow definition language, it completely handles WPDL as well as XPDL, and also it is able to support the Wf-XML 2.0 interoperability interface, which has recently released by WfMC, for interacting with other heterogeneous workflow systems. Our focus is on the workflow enactment component (the so-called workflow engine).

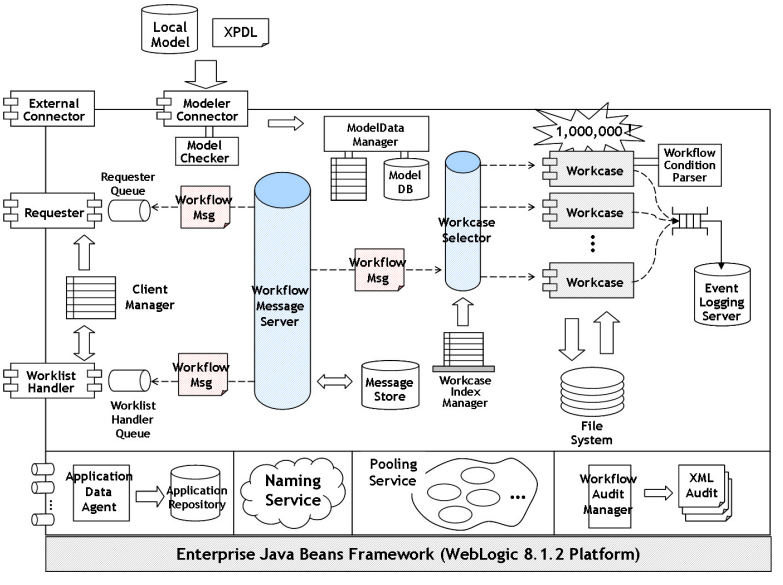


Fig. 3. The Overall System Architecture of e-Chautauqua

The overall system architecture of e-Chautauqua as shown in Fig. 3 consists of several agent-based components, such as PDL (process definition language) Agent, OM (organization management) Agent, Administration Agent, Tool Agent, requester, worklist handler, script interpreter, logger, workflow Message Beans using JMS message queue, and Enactment & Control Agent consisting of workcase objects registered in the workcase pool. These all components are deployed on an Enterprize Jave Beans middleware platform[3] and they asynchronously communicate with each other through a JMS-based message queue mechanism named the workflow message beans.

[3] Note that we used the Weblogic 8.1.2 package, but it won't be dependent on a specific EJB package. Recently we have successfully deployed our system on JBoss enterprize middleware platform, too.

The core component that distinguishes e-Chautauqua from other workflow engines is the workcase object component. That is, as stated in the conceptual architecture section, almost all conventional workflow engines are based on the activity-oriented workflow architectural style, but on the other hand e-Chautuaqua's engine is based on the workcase-oriented workflow architectural style. In the activity-oriented workflow engines, the control flow management functionality is performed through interactions among activity objects, because the core component is the activity object component. In contrast to this, the control flow management is done by the workcase objects in the workcase-oriented workflow engine (e-Chautauqua), and so the activity precedence information is stored to the inside of each workcase object as data. Conclusively, the most valuable benefit that we expect from the workcase object component is that the control flow management functionality can be done by much simpler mechanism, and also it can be efficiently done with much smaller amount of computing resources, because the workcase-oriented workflow engine is able to dramatically decrease the number of objects resided and managed in the system. These facts mean also that the workcase-oriented workflow engine like e-Chautuaqua must be more appropriate and durable for the very large scale workflows, and it must give satisfaction in the higher degree of workflow instantiation complexity.

3.2 Implementation of e-Chautauqua System

In this section, we present a real appearance of the e-Chautauqua workflow management system through two operational examples — buildtime client and runtime client.

3.2.1 e-Chautauqua Buildtime Client: The ICN-based Workflow Modeler

Fig. 4 is to show a captured screen of the e-Chautauqua buildtime client that is the modeling result of the highway and road maintenance process, as a practical very large scale workflow model, that is currently operated in a nation-wide large scale enterprize (Highway Construction Company) in South Korea. About 100 activities including compound activities of subworkflows are associated with the workflow model. In graphical notations of the workflow model, the small dotted circles represent conjunctive (or parallel) control logics - *AND-split* and *AND-join nodes*. Also, the company expects that the number of workcases (or instances) spawned from the workflow model is more than several hundred-thousands workcases because of a lot of road maintenance cases requested from several hundred branch offices geographically scattered on the whole country. We strongly believe that this model is a typical example of maximally parallel and very large scale workflow model, and it necessarily requires the fundamental and architectural renovation of the traditional workflow management systems in order to not only efficiently handle but also maximally utilize the company-wide computing resources. Based upon consideration of these situations, the e-Chautauqua system ought to be the impeccable solution as an underlining infrastructure for maximally parallel and very large scale workflow applications.

3.2.2 e-Chautauqua Runtime Client

In this section, we present an operational example of e-Chautauqua runtime client by showing captured screens of the runtime client user interface and its related functionalities. In Fig. 5, The first screen from the lowest layer in the overlapped screens is the

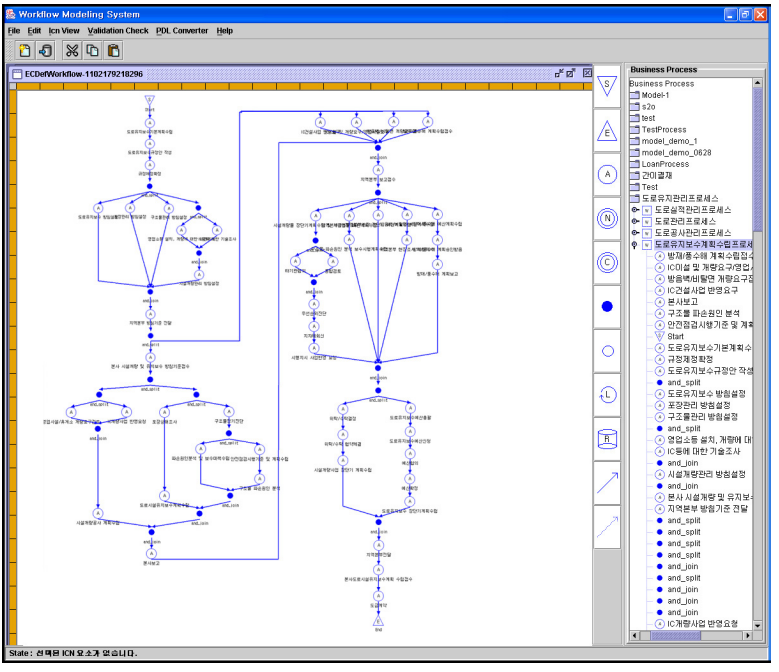


Fig. 4. A Modeling Example of the Very Large Scale Workflow Model

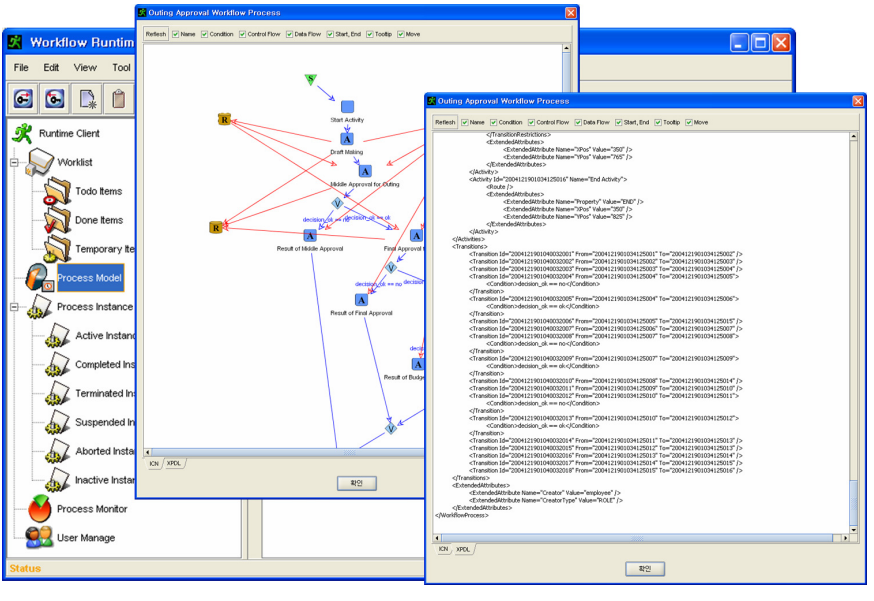


Fig. 5. e-Chautauqua's Runtime Client

runtime client's user interface having a worklist folder, workflow (or process) model folder, running workcases (instances) folder, and a workflow monitoring folder. The second screen is showing a workflow model graphically represented by the information control net, and on the top of the screen there are several toggle switches used for alternatively showing the workflow model information such as activity's name, transition informations, control flow, data flow and so on. The last third screen shows the workflow model of the second screen represented in XPDL.

Summarily speaking, let us briefly describe a little more details of the components. The workflow process (procedure) modeler and simulator are also responsible for describing, specifying, and simulating workflow procedures through the graphical user interfaces. The graphical description of a workflow procedure can be represented in a textual form through the workflow procedure description language (WPDL) or the XML workflow procedure description language (XPDL). The workflow administration and monitoring services are used for managing the administrative tasks, like initiating workcases, performing dynamic changes of data-flow and control-flow in a workcase, terminating or holding tasks of a workcase, monitoring the execution statuses of tasks associated with a workcase, and serving other administrative and monitoring requests. The services are operable on a graphical user interface, too. The script interpreter performs the definition of transition conditions in buildtime and their processing during runtime.

4 Conclusion

In this paper, we have newly defined the distribution factors and the conceptual framework for guiding the design of workflow architectures. Based on the framework, we proposed the workcase-based distributed workflow architecture for the very large scale workflow applications, and implemented the architecture by using an enterprize java beans framework and named it e-Chautauqua workflow management system. Finally, we have presented the practical examples—Buildtime client (ICN-based workflow modeling tool) and Runtime client— and confidently reached at the decision that the architecture and system should be affordable for the very large scale workflow requirements. Based upon the results of this paper, we would have a great chance for the system to be extended as as to become a commercialized system.

Acknowledgement. The research was partially supported by the special BEIT research fund of Kyonggi University. And the research result is based upon the BEIT's Grid Computing facility.

References

1. Alonso, D., et.al.: Failure Handling in Large Scale Workflow Management Systems, IBM Research Report RJ9913 (November 1994)
2. Alonso, D., et.al.: Exotica/FMQM: A Persistent Message Based Architecture for Distributed Workflow Management. In: Proceedings of the IFIPS Working Conference on Information Systems for Decentralized Organizations (August 1995)

3. Mani Chandy, K., Neuse, D.: Linearizer: A Heuristic Algorithm for Queuing Network Models of Computing Systems. Communication of the ACM 25(2) (1982)
4. Ellis, C.A., Morris, P.: The Information Control Nets Model. Performance Evaluation Review 8(3) (1979)
5. Ellis, C.: Team Automata. In: Proceedings of the ACM Group'97 Conference (November 1997)
6. Ellis, C., Maltzahn, C.: Chautauqua: A Flexible Workflow System," Proceedings of the 30th HICSS Conference (January 1997)
7. Ellis, C., Kim, K.: A Framework and Taxonomy for Workflow Architectures. In: The Fourth International Conference on Design for Cooperative Systems (May 2000)
8. Jablonski, S., Bussler, C.: MOBILE: A Modular Workflow Model and Architecture, University of Erlangen Internal Report (1995)
9. Kim, K.: Practical Experiences and Requirements on Workflow. In: Conen, W. (ed.) Coordination Technology for Collaborative Applications. LNCS, vol. 1364, pp. 145–160. Springer, Heidelberg (1998)
10. Kim, K., Ellis, C.: Performance Analytic Models and Analyses for Workflow Architectures. Information Systems Frontiers: A Journal 3(3), 339–355 (2001)
11. Miller, A., et.al.: CORBA Based Run Time Architectures for Workflow Management Systems, LSDIS Internal Lab. Report, University of Georgia (1996)
12. Perry, D., Wolf, A.: Foundations for the Study of Software Architectures. ACM SIGSOFT Software Engineering Notes 17(4) (1992)
13. Podgurski, A., Clarke, L.A.: A Formal Model of Program Dependencies and Its Implications for Software Testing, Debugging, and Maintenance. IEEE Trans. on Software Engineering 16(9) (1990)
14. Rolia, J.A.: Predicting the Performance of Software Systems", Technical Report CSRI-260, Computer Science Research Institute, University of Toronto (January 1992)
15. Rolia, J.A., Sevcik, K.C.: The Method of Layers. IEEE Transaction On Software Engineering 21(8) (1995)
16. Wallnau, K., Long, F., Earl, A.: Toward a Distributed Mediated Architecture for Enterprise-wide Workflow Management. In: Proceedings of the NSF Workshop on Workflow and Process Automation (May 1996)
17. Weissenfels, J., et.al.: An Overview of the Mentor Architecture for Enterprise Wide Workflow Management. In: Proceedings of the NSF Workshop on Workflow and Process Automation (May 1996)
18. Ellis, C.A., Nutt, G.J.: Process Models and Systems for Logistics Support, A Proposal to the US DOD ARPA, CU Proposal No. 0595.00.0738B
19. Palaniswami, J.L., Shevchenko, I., Mattie, A., Reed-Forquet, L.: Web-based Multi-Paradigm Workflow Automation for Efficient Health-care Delivery. In:Proceedings of NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-Art and Future Directions (May 1996)
20. Chazin, D.: An Architecture and Transaction Model for Large Federated Databases, Ph. D. Thesis, University of Illinois at Urbana-Champaign (May 1996)
21. Alonso, G., Schek, H.-J.: Research Issues in Large Workflow Management Systems. In: Proceedings of NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-Art and Future Directions (May 1996)

Building Web Application Fragments Using Presentation Framework

Junghwa Chae

École Polytechnique de Montréal
Montréal, Québec, Canada
junghwa.chae@polymtl.ca

Abstract. Building web applications is a complex and time-consuming process, especially when processes are synchronous. In this paper, we propose a role-based presentation framework for asynchronous message delivery of web components. This asynchronous framework clearly separates the role of elements involved in the presentation layer of web applications and defines their relationships. Essentially, this framework is based on two design patterns and one programming model: the model-view-controller design pattern, Real-Time Messenger design pattern, and JMS programming model.

1 Introduction

The web has become the global infrastructure for delivering information and services [11]. The demands and expectations of information providers and consumers are pushing web technology towards a higher level of presentation quality, including active contents [17]. This rapid emergence of web-based information systems in recent years presents a serious challenge to the development skills of information technology professionals. Until recently, many such systems were simple and stand-alone applications. Now, they have become extensive and complex e-commerce applications. Accepting fast changing enterprises and consumer demands such as those for the purpose of developing high-quality web applications requires highly-skilled IT experts with multiple web-based expertise in JSP and Java servers [16]. Another aspect to the complexity of web applications is that application development relies heavily on the knowledge and experience of individual developers and their individual development practices rather than on standardized practices [16,12,8]. Therefore, developers need new techniques for developing web applications within a framework.

In this paper, we suggest the presentation framework that is a new approach which provides a clear definition of the entities in a presentation layer and their relationships based on their role. We propose a framework focused mainly on asynchronous message delivery of web components. This framework basically follows the well-known model-view-controller (MVC [9,15]) design pattern fit for the architecture of an interactive system. It is also an applied Java Message Service (JMS [13]) to support asynchronous message delivery of web components.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 929–940, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Likewise, to improve the efficiency of the system when delivering messages between web components, we take advantage of the Real-Time Messenger design pattern [14]. This paper focuses on the presentation layer of a web framework. It applies the mechanism of asynchronous message delivery and design patterns suitable for the presentation layer. This work helps save developers time in building basic functionality. Likewise, this allows the creation of customizable and selectable framework solutions for different business requirements.

The rest of this paper is organized as follows: Section 2 describes the purpose in detail; Section 3 outlines the basic strategy used in the suggested framework; Section 4 discusses the design of the presentation framework; Section 5 describes related works; and Section 6 discusses the conclusion with future work.

2 Purpose

Most web applications are divided into three logical layers: business, data, and presentation [10]. The presentation layer forms presentation logic by accepting demands from clients such as HTML, Applet, and XML, and responding appropriately. For dynamic web contents, it uses web components such as Java servlets and Java server pages (JSP). These two layer-based programs can be developed separately by programmers because each layer can be executed in different web environments. Although efforts to make components of the business layer [1], which expresses the business logic, have come a long way, there have been few studies done about the presentation layer consisting of the screen and the handling of the clients request. Therefore, we conclude that developers need new techniques to develop the presentation layer of a web application in a web framework.

The benefits of a well-defined and supported software process are well known [4]. For these reasons, we propose to break down the framework into the number of processors, based on their role in the presentation layer. The framework of requirements can be derived by examining the following founding principles of software engineering [1]:

- ***Formalization:*** A precise definition of the characteristics and relationships of the different elements of a presentation which supports developers understanding of the application at hand.
- ***Modularization:*** Modularization mechanisms could be used to divide a complex problem into smaller, simpler components.
- ***Abstraction:*** The web implementation technology should abstract away from low level, unimportant details, identifying important concepts and relationships.
- ***Customization:*** The web implementation technology should provide general mechanisms that support the development.

In the past, computer systems have been completely contained within a single organization. In traditional synchronous RPC calls, the sender waits for the reply from the receiver before moving on, while other operations are being blocked

until the call returns. It is synchronous at the clients end. For this synchronous system, the processing is suspended until the client receives a response from the server. Thus, it ties up the resources of the server since it cannot handle other processes until a reply is returned. This results in wasted resources and causes user misunderstanding, with the application shown as being down because the GUI screen application is incapable of responding.

Asynchronous message delivery [13] means that the client sender conducts other processes instantly. Therefore, the sender no longer needs to wait for a response from the receiver before dealing with other processes after making a request. When this is applied to the message delivery of web components, the sender is able to cope with other operations. In recent years, the web has become a general IT platform, and complex and large-scale distributed systems are widely used. Although synchronous message delivery is sufficient in cases of small traditional applications, the ideal method of message delivery should be not only synchronous, but also asynchronous.

There are several advantages of asynchronous message delivery. Time-wise, components that communicate with one another do not all need to be available at the same time. Position-wise, the position between the component sending the real-time message and the component receiving the message can be changed, since the sender and receiver become separated. In the case where the computer of the component is out of order, the component within the computer can move to other functional computers for activation. Therefore, the application allows for a certain amount of defects. Since the message sender or message receiver all become separated, the user interface allows other processes to progress when delivering the message or sending the request. This enables the creation of user-friendly applications. Given these advantages, we decided to introduce a new asynchronous approach for the presentation layer of web applications. In many cases, the look of applications is presented in user interfaces irrespective of data structures and business logic [15]. There are two patterns that can be used for viewing information. The Observer design pattern [3] is generally used to design web components that communicate with other web components, i.e., view to view. The Observer pattern is mainly used to notify each presentation layer that state information has changed in the model. In view to view, it is desirable to direct the model information that is not used when giving and taking the revised matters and interactive orders firsthand. Thus, in case of communication between web components, Real-Time Messenger design pattern could be used.

3 Basic Strategies

There are three basic strategies used in this study. First, the MVC pattern is used in web applications in order to separate business logic and data from presentation logic. The basic architecture of this framework is based on the MVC design pattern. Second, the JMS API is used for asynchronous messages. Third,

Real-Time Messenger design pattern is used to achieve system efficiency in delivering the message between view objects. These vital strategies are described below:

3.1 The Model-View-Controller (MVC) Design Pattern

Web applications, like other interactive software systems, can benefit by being architected with the MVC design pattern [15]. In J2EE-based web applications, this approach uses Enterprise JavaBeans (EJBs [1]) as the model module. It constructs a view using HTML and JSP and implements the controllers using servlets and session EJBs. Thus, even though the model, view, and controller performances are definitely separated, they should maintain their constant relations. Figure 1 shows the architecture of MVC.

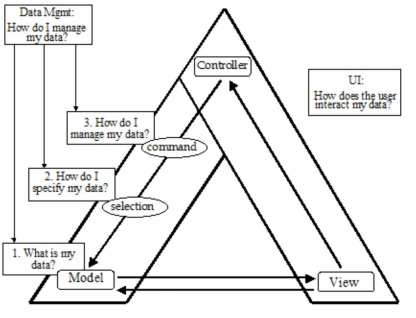


Fig. 1. MVC architecture

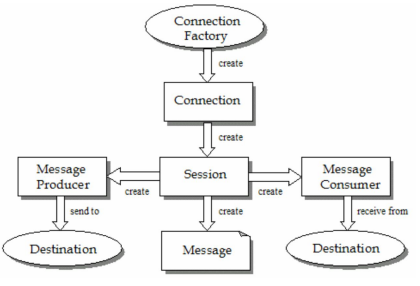


Fig. 2. JMS API programming model

3.2 JMS

JMS APIs most important purpose is to allow Java applications to access existing Message-Oriented Middleware systems [13]. JMS also provides an asynchronous request/reply model. JMS is used for loosely coupled asynchronous communications.

The JMS API is an integral part of the J2EE platform. Application developers can use messaging with components using JMS API. JMS API allows the different components to interact with each other efficiently without tying up the network or other resources. Therefore, JMS can be used as an alternative synchronous solution. Figure 2 shows the structure of the JMS API programming model.

Likewise, it offers common functionality for the presentation of web applications using Java Message Service and Real-Time Messenger design pattern. Java Message Service can be used as an alternative synchronous solution, making use of the asynchronous message delivery mechanism. The Real-Time Messenger design pattern presents a structurally organized message-passing process between views under the Internet development environment.

3.3 Real-Time Messenger Design Pattern

The key idea of the Real-Time Messenger design pattern is to separate the methodology of design from the implementation according to the messages way of delivery. It is therefore an efficient way to reduce unnecessary coupling by means of a view message delivery to the model.

The interior processing structure system, which applies the established MVC architecture, causes unnecessary coherence on account of the message delivery system of view via the model. However, the observer pattern is applied to the revised data of model module and the Real-Time Messenger design pattern when conveying the interactive message between view objects; thus beefing up the efficiency of the message processing. Figure 3 illustrates the class diagram of the Real-Time Messenger design pattern.

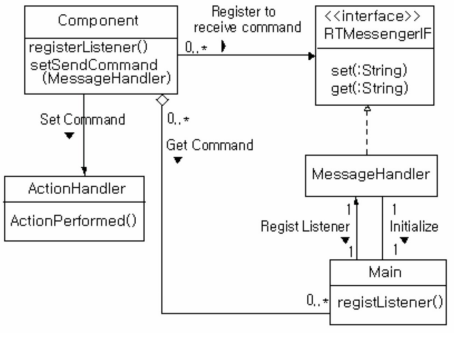


Fig. 3. Class diagram of Real-time messenger design pattern

4 Architecture

The proposed presentation framework is organized according to activities of presentation layer in a number of inter-dependent tasks, as shown in Figure 4. Figure 5 shows the proposed context diagram of the framework. This is essentially based on the MVC paradigm. Here, the MVCs view and controller are combined into a presentation layer. In MVC, the view displays information to the user.

The proposed presentation framework clearly separates the role of elements involved in the presentation layer and it defines their relationships. It consists of the following seven components: RequestController, ActivityProcessor, MessageProcessor, ScreenFlowController, ScreenPrecessor, and BusinessLogicController.

The seven components are grouped according to the MVC design pattern. The view module is seen through the ScreenProcessor. The view module interacts with standard data access technology, including EJBs, JDBC, and JNDI. The controller module consists of an ActivityProcessor, EventProcessor, MessageProcessor, and ScreenFlowController.

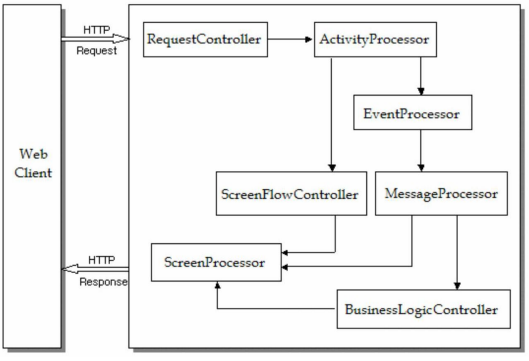


Fig. 4. Presentation framework architecture

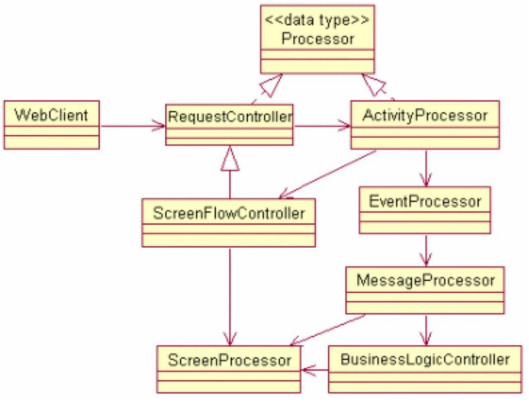


Fig. 5. Context diagram

These are defined as the standard interfaces communicating with the MVCs model part. The presentation layer interacts closely with the application servers. So, the BusinessLogicController is used to choose the model module, i.e., it sends data to an appropriate business bean in the business layer. Each components has distinguishing features on their roles described below:

The view, together with the controller, which processes the users interaction, comprises the applications user interface. The model is the portion of the application that contains both the information represented by the view and by the logic that changes this information in response to user interaction. This framework is chiefly made up of view and controller since it is the subject of the presentation layer.

(1) RequestController

The RequestController of this framework activates a servlet, analyzing the URI in the request of the client. This is a technology used when accepting requests

at the presentation layer of web applications. A servlet is capable of activating another servlet using URI. This is the role of the servlet engine which manages the servlets. Thus, the user request is sent to the ActivityProcessor or ScreenProcessor through the URI.

(2) ActivityProcessor

The ActivityProcessor, which plays the role of the front controller, initiates the context to be used in the EventProcessor and ScreenFlowController. When the client sends a request, it analyzes the URI and passes the screen property to ScreenFlowController. Table 1 shows the flow of the ActivityProcessor.

Table 1. Flow of ActivityProcessor

a. Initiate context to be used in the EventProcessor and ScreenFlowController.
b. Instantiate EventProcessor.
c. Instantiate ScreenFlowController.
d. Pass the screen property to ScreenFlowController.

(3) EventProcessor

The EventProcessor handles the 'HttpRequest' event using the URI which is analyzed by the ActivityProcessor. The analyzed URI is used to convert the parameter into the event such as "page?parameter=value" or "page?parameter". Table 2 shows the flow of the EventProcessor.

Table 2. Flow of EventProcessor

a. Acquire the request object from the pool, or initiate it if there is no object.
b. Select a screen name.
c. Select activity to be dealt with based on the URI.
d. Instantiate BusinessLogicController to handle the event from the web client.
e. Define the event dimension using the URI.
f. Generate events for event dimension in the web tier.
g. Transfer the event by way of the MessageProcessor.

(4) MessageProcessor

The MessageProcessor is the key, since it is the controller of the framework. The MessageProcessor is used to deliver and handle the message between the web components. It is designed to asynchronously deal with message of the web components. In this framework, the MessageProcessor using JMS plays the role of message producer in JMS models. It can also be seen as a client form, i.e., web components get connected after seeking connection factories.

Table 3. Flow of MessageProcessor

a. Identify the property of an event which is delivered from EventProcessor. **b.** Send the message to the BusinessLogic-Controller or the ScreenFlowController based on the property of the event.

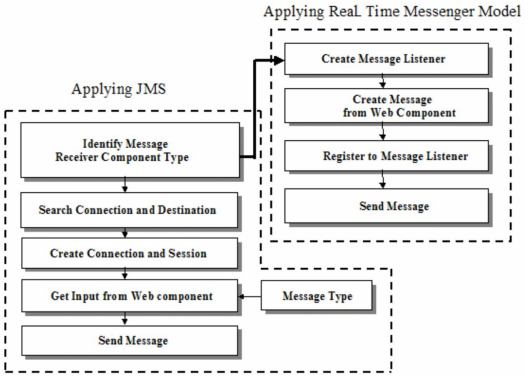


Fig. 6. Structure of MessageProcessor

They then send and receive messages through the message server, elucidating particular messages. This message controller is only in the form of a client which connects to a JMS destination. Table 3 shows the flow of the MessageProcessor.

The structure of the message controller is illustrated in Figure 6. Figure 6 is classified into two types according to the ways of message delivery of components. The first, applying JMS, is the direct message delivery in the view to view object. In the previous design ways, the Observer design pattern is mainly used to deliver the changed values to the presentation layer from the business layer. However, in this case, direct distribution of changed values is desirable in the view to view object, since the model information is not used. The second, the Real-Time Messenger design pattern, is designed to convey at real-time, instead of being conveyed through the model. The Real-Time Messenger pattern has the component that implements the Real-Time Messenger interface. The Real-Time Messenger interface, in turn, has the action handler.

This component contains the method that can register the listener and send the message. The component sends the message, initiating the main program with the message listener, which in turn initiates the listener to deal with it. In this framework, the Real-Time Messenger pattern is used to directly deliver the message between web components. Table 4 defines the Interface of the Message-Processor.

The JMS model deals with the event and delivers the message via the component in the business layer. The JMS model is designed with the ability to connect to JMS server to assist in asynchronous message delivery. For the client

Table 4. Interface of MessageProcessor

```
public interface MessageProcessor{
    public abstract void analyze MessageReceiver(MessageReceiver receiver);
    public abstract setMSEnv();
    public abstract void porcessMessage(WebComponent web);
    public abstract void sendMessage(); }
public interface MessageReceiver {public abstract void receiveMessage(); }
public interface WebMessageReceiver extends MessageReceiver {}
public interface BusinessMessageReceiveer entends MessageReceiver {}
```

request, the model should wait for the response (after adequate processing) from the business logic concerning the request which was addressed. It is allowed to convert it into the form of event. It is also allowed to send the message by using JMS for asynchronous delivery message support.

(5) ScreenFlowController

The ScreenFlowController controls the flow of the change in case the model information or the screen is modified. Figure 5 shows flow of the ScreenFlow-Processor.

Table 5. Flow of ScreenFlowController

a. Acquire the ScreenFlowController; instantiate if there is no instance.
b. Manage the defined screen using the hash map.
c. Map the screen suited for the situation.
d. Pass the control by using the form used in the ScreenProcessor.

(6) ScreenProcessor

In general, the view is shown as components such as HTML pages, JSP pages, or servlets. The ScreenProcessor takes the role of the view in this framework.

Table 6. Flow of ScreenProcessor

a. Obtain screen name from the RequestController, ScreenFlowController, or BusinessLogicController. The current URI shall be stored in the request object created by the RequestController.
b. Save screen name obtained by analyzing URI stored in the request object or originating from the ScreenFlowController.
c. Choose screen type suitable for the screen name.
d. Send response for the layout of screen.

The view component is seen through the ScreenProcessor, i.e., the screen is created by the contents of screen definition declaratively composed in line with the request object made by the ScreenFlowController, the MessageProcessor, or the BusinessLogicController. A created screen-type file is the layout of the screen. Figure 6 shows flow of the ScreenProcessor.

(7) BusinessLogicController

The role of the BusinessLogicController plays is to choose the model module that will handle the message. The interaction between the business objects and the presentation objects is managed through the BusinessLogicController. This provides methods that allow the business objects to interact with the presentation layer. The MVCs model module is embodied without reference to the information of any presentation layer. The position of the model part which actually handles the data is put in different layers based on the size of application. In case of light-weight applications, the model module will be put in a web layer instead of a business layer. In case of middle or heavy-weight applications, a business bean, which has the role of a model, will be deployed in a business layer. The developer can decide the way by moderately mixing these two kinds ways with the size of web applications. In this framework, the developer only uses the same interface irrespective of the model module position regarding the method of business logic processing. That is, the framework provides the position change of model as a parameter; thus paving the way for adequate revision according to the size of web applications.

5 Related Works

Most of the presentation layer frameworks currently offered take on various tastes including Suns JSP Model 2 paradigm (Model 2 [4]). Struts [3] is the framework with Model 2 architecture based on the MVC design pattern. This focuses on controlling the flow for JSP-based web applications. Turbine [4], a similar framework based on servlets, is nicknamed as Model 2+1 aimed at making the gathering of components reusable to develop an easy server-side application. It also provides the same coherence as template engines such as webMacro [6], Freemarker and Velocity [5].

The presentation layer created by Barracuda uses the method of template engine programming, used by servlets. Most of the presentation frameworks above perform the function of Model 2. Model 2 is an approach that shows MVC patterns such as Sun showing web applications focusing on JSP form. The presentation framework is therefore originally based on the MVC design pattern. In [15], they use the MVC design pattern as a way to more flexibly partition web applications in development. Likewise, [16] mentions three problems in developing the web applications – the lack of experts, the problem of maintenance, repair and expansion, and development time. A general presentation framework is therefore recommended. The Roja framework [2] has provided solutions capable of developing lightweight, middleweight, and heavyweight e-commerce in

accordance with the complexity of the system and the demand standards of business. [7] proposes a systematic and flexible scripting framework for fast web applications development.

Not all applications will benefit from the same level of framework development. The framework suggested in this paper has several characteristics. First, it clearly separates the role of elements involved in the presentation layer and defines their flows. We applied the well-defined founding principles of software engineering for the design framework. Second, the architecture follows the MVC structure like most of the frameworks recommended earlier. The benefit is that the presentation logic can be completely separated from the application logic and business logic. However, there is a weakness associated with the MVC pattern. We used a design pattern to overcome the demerit of it. In particular, it is capable of asynchronous message processing of web components presentation layer.

6 Conclusion and Future Work

This paper suggested a role-based presentation framework that concentrates on asynchronous message delivery of web components. We defined the overall architecture as the flow of constituents to establish the concept and specified the interface of each module in detail. To design the framework, the mechanism of asynchronous message delivery and design patterns have been described. This framework provides a flexible and systematic function to develop more standardized web applications faster. This encapsulation of the framework also allows presentation logic developers to focus on the main development tasks without being too concerned that another business logic developer will misuse its object. This framework is still in its early stages; thus a lot of work still remains to be done. We plan to further develop the presentation framework based on the architecture of this paper. Likewise, we also plan to develop actual web applications in the framework.

Acknowledgments. The research of Junghwa Chae was supported by Institute for Information Technology Advancement (IITA) & Ministry of Information and Communication (MIC), Republic of Korea.

References

1. Enterprise JavaBeans specifications, `http://java.sun.com/products/ejb/docs.html`
2. Roja framework and sample project, `http://rojaware.wo.to`
3. Struts. `http://jakarta.apache.org/struts/index.html`
4. Turbine:Open source presentation framework. `http://jakarta.apache.org/turbine`
5. Velocity: Open source presentation framework, `http://jakarta.apache.org/velocity`

6. Web Macro: Open source presentation framework. http://webmacro.org
7. Web service user interface WSUI 1.0
8. Bochicchio, M.A., Paiano, R.: Software reuse in hypermedia applications. In: Khosrowpour, M. (ed.) IRMA Conference, pp. 826–829. IDEA Group Publishing, USA (2000)
9. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Pattern-Oriented Software Architecture: A System of Patterns. John Wiley & Sons, New York (1996)
10. Conallen, J.: Modeling web application architectures with UML. Commun. ACM 42(10), 63–70 (1999)
11. Gellersen, H.W., Gaedke, M.: Object-oriented web application development. IEEE Internet Computing 3(1), 60–68 (1999)
12. Ginige, A., Murugesan, D.: Web Engineering: An Introduction. University of Western Sydney, Australia (2000)
13. Giotta, P., et al.: Professional JMS programming. Xerox_Press (2001)
14. Kim, Y.J.: Design pattern for message passing between function component:real-time messenger pattern. Master's thesis, Chonbuk University, Korea (2000)
15. Leff, A., Rayfield, J.T.: Web-application development using the model/view/controller design pattern. In: EDOC, pp. 118–127. IEEE Computer Society Press, Los Alamitos (2001)
16. Saimi, A., Syomura, T., Suganuma, H., Ishida, I.: Presentation layer framework of web application systems with server-side java technology. In: COMPSAC, pp. 473–478. IEEE Computer Society Press, Los Alamitos (2000)
17. Taylor, C.: Applying software engineering principles to a web site development process. Master's thesis, DePaul University (2000)

Three–Dimensional Bursting Simulation on Two Parallel Systems*

S. Tabik[1], L.F. Romero[2], E.M. Garzón[1], I. García[1], and J.I. Ramos[3]

[1] Departamento de Arquitectura de Computadores y Electrónica, Universidad de Almería, 04120 Spain
[2] Departamento de Arquitectura de Computadores,Complejo Politécnico, Campus de Teatinos, Apdo. 4114, 29080 Málaga, Spain
[3] Room I–320–D, E.T.S. Ingenieros Industriales, Universidad de Málaga, Plaza El Ejido, s/n, 29013 Málaga, Spain
jirs@lcc.uma.es

Abstract. This work proposes a mathematical model and its parallel implementation on two different computer architectures to simulate three–dimensional bursting phenomena. The mathematical model consists of four, nonlinearly coupled partial differential equations and includes fast and slow subsystems. The differential equations have been discretized by means of a linearly–implicit finite difference method in equally–spaced grids. The resulting system of linear algebraic equations at each time level has been solved by means of the Generalized Minimal Residual (GMRES) method with Jacobi preconditioning. The parallel implementation has been formulated using the message passing paradigm, where the linear system of equations has been solved by means of the parallel sparse solver GMRES included in the portable extensible toolkit for scientific computation (PETSc). The parallel simulation has been evaluated on (1) A cluster of biprocessors Xeon(TM) and (2) A SGI Altix 3700 Bx2. Similar performance has been obtained on both platforms, with better scalability on the DSM multiprocessors.

Keywords: Bursting; quenching; MPI; GMRES; DSM.

1 Introduction

Bursting phenomena occurs in physiology and biology. For example, neurons communicate by firing and transmitting action potentials. Commonly, action potentials occur in a periodic fashion, as in response to a constant applied current of sufficiently large magnitude. In addition, many cell types, e.g., pancreatic β-cells, exhibit more complex behavior characterized by brief burst of oscillatory activity interspersed with quiescent periods during which the cell membrane potential changes slowly. Bursting is a phenomenon characterized by brief bursts of

* This work was supported by the Ministerio de Educación y Ciencia of Spain under Projects TIN2005-00447, FIS2005–03191 and Fondos FEDER.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 941–949, 2007.
© Springer-Verlag Berlin Heidelberg 2007

oscillations interspersed with quiescent periods. Models of bursting electrical activity can be classified into two main groups. The first and earliest one was based on the assumption that bursting was caused by an underlying slow oscillation in the intracellular Ca^{2+} concentration [1,2]; however, recent experiments indicate that this assumption is not entirely correct and, as a consequence, models relying on alternative mechanisms have been developed [3].

The different known bursting mechanisms can be classified into three main groups. In type I, burst arise from hysteresis and bistability as in the pancreatic β–cell model. In type II, bursts arise from an underlying slow oscillation, while, in type III, bursting arises from a subcritical Hopf bifurcation [1,2,4,5]. This classification is by no means complete, for it is based on knowledge acquired with nonlinear ordinary differential equations. In extended systems, i.e., spatio–temporal systems, one can easily show that bursting occurs through one of the three types mentioned above provided that the system is homogeneous. However, when the system is not initially homogeneous and the diffusion coefficients are not nil, the homogeneous bursting solution may bifurcate and result in spatio–temporal nonhomogeneities and quenching.

The objective of this paper is several–fold. First, a numerical technique based on the solution of a nonlinearly coupled, four–equation is proposed to describe a three–dimensional model of bursting. This model has been selected so that it exhibits bursting under homogeneous conditions, and both bursting and quenching for non–homogeneous initial conditions. The source or reaction terms of these partial differential equations can be classified into two subsystems exhibiting fast and slow behavior; the accurate simulation of the fast processes demand the use of small time steps, while the extended system considered here requires the use of sufficiently small spatial steps in order to accurately resolve the steep gradients of the dependent variables.

Second, a parallel implementation of the three–dimensional bursting model is developed using the message passing approach, where all communications between processors are established via MPI. Besides, the parallel solution of the linear system of equations is carried out by means of GMRES solver included in PETSc. The parallel simulation is evaluated on two parallel architectures: (1) A cluster of five Intel(R) Xeon(TM) biprocessors and (2) A SGI Altix 3700 Bx2 of 8 processors. The obtained results are discussed and new proposals are made to improve the performance on the DSM platform. The proposed parallel approach and proposals can be valuable for other SPMD applications that are based on the message passing paradigm and that include the solution of sparse system of equations similar to that described in this work.

2 Formulation and Numerical Method

Bursting in extended systems has been simulated by means of the following nonlinearly coupled system of partial differential equations

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{D}\left(\frac{\partial \mathbf{U}^2}{\partial x^2} + \frac{\partial \mathbf{U}^2}{\partial y^2} + \frac{\partial \mathbf{U}^2}{\partial z^2}\right) + \mathbf{S}(\mathbf{U}) \tag{1}$$

where $\mathbf{U} = (u, v, w, p)^T$, $\mathbf{D}$ is a diagonal matrix with components equal to D_u, D_v, D_w and D_p, t is time, x, y and z are Cartesian coordinates and $\mathbf{S} = (S_u, S_v, S_w, S_p)^T$ is the following column vector of the source/reaction terms

$$\begin{cases} S_u = f(u) - v - gw(u - u_0) \\ S_v = \frac{1}{5}(v_\infty(u) - v) \\ S_w = f_w(w) + \alpha_w(p - 0.3) \\ S_p = \beta_p((1 - p)H(u) - p) \end{cases} \tag{2}$$

where

$$\begin{cases} f(u) = 1.35u(1 - \frac{u^2}{3}) \\ f_w(w) = -0.2(w - 0.2)(w - 0.135)(w - 0.21) \\ v_\infty(u) = tanh(5u) \\ H(u) = \frac{3}{2}(1 + tanh(5u - 2.5)) \end{cases} \tag{3}$$

with $u_0 = 2$, $\alpha_\omega = 0.002$, $\beta_z = 0.00025$ and $g = 0.73$. The functions $f(u)$ and $f_\omega(\omega)$ have three zeros each, whereas $v_\infty(u)$ and $H(u)$ are monotonically increasing functions of their arguments.

Eq. 1 is a simplified model of bursting electrical activity in cells, where u and w are the fast system and represent the currents of activated and voltage–dependent channels, whereas w and p constitute the slow subsystem and represent a current and its activation

Eq. 1 have been solved in a parallelepiped $\Omega \equiv [-L_x, L_x] \times [-L_y, L_y] \times [-L_z, L_z]$ and homogeneous Neumann boundary conditions have been used on all the boundaries. The initial conditions used in the computations are

$$u = -2.5, v = -0.2, \omega = -0.5, z = 0.5 \text{ in } \omega$$
$$u = 2.5, v = -0.2, \omega = 0.5, z = -0.5 \text{ in } \Omega - \omega$$

where $\omega = [-a, a] \times [-b, b] \times [-c, c]$, $a = \frac{L_x}{2}$, $b = \frac{L_y}{2}$ and $c = \frac{L_z}{2}$. These initial conditions were selected so that, under homogeneous conditions, Eq. 1 exhibits bursting oscillations.

By discretizing the time variable in Eq. 1 by means of (second–order accurate) Crank-Nicolson technique one can obtain a nonlinear system of elliptic equations at each time step. These elliptic equations can be linearized with respect to time in order to obtain a system of linear elliptic equations at each time step, i.e., the nonlinear terms $\mathbf{S}^{n+1}$ was approximated by means of the second-order accurate term $\mathbf{S^n} + \mathbf{J}^n \Delta\mathbf{U}$ where $\mathbf{J} = \frac{\partial \mathbf{J}}{\partial \mathbf{U}}$ denotes the Jacobian of the source terms, the superscript n denotes the n–th time level, i.e., $t^n = n\Delta t$, $n = 1, 2, 3, \ldots$ where Δt is the time step, and $\Delta\mathbf{U} \equiv \mathbf{U}^{n+1} - \mathbf{U}^n$. The resulting system of linear elliptic equations was discretized in space by means of second–order accurate central finite difference formula in a regular grid of $N_x \times N_y \times N_z$ points, and the resulting system of linear algebraic equations can be written as

$$\begin{aligned} &(\mathbf{I} - \tfrac{k}{2}\mathbf{J}^n_{i,j,k})\Delta\mathbf{U}_{i,j,k} \\ &-\alpha_x(\Delta\mathbf{U}_{i+1,j,k} - 2\Delta\mathbf{U}_{i,j,k} + \Delta\mathbf{U}_{i-1,j,k}) \\ &-\alpha_y(\Delta\mathbf{U}_{i,j+1,k} - 2\Delta\mathbf{U}_{i,j,k} + \Delta\mathbf{U}_{i,j-1,k}) \\ &-\alpha_z(\Delta\mathbf{U}_{i,j,k+1} - 2\Delta\mathbf{U}_{i,j,k} + \Delta\mathbf{U}_{i,j,k-1}) = \mathbf{T}^n_{i,j,k} \end{aligned} \tag{4}$$

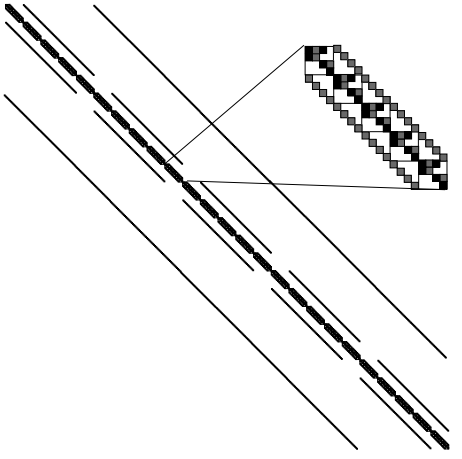


Fig. 1. Matrix pattern corresponding to a $5 \times 5 \times 5$–point grid, where, grey points present nonzero time-independent elements and black points present nonzero elements that have to be updated at each time step

where $\mathbf{I}$ is the identity or unit matrix, $\alpha_x = \frac{k}{2\Delta x^2}\mathbf{D}$, $\alpha_y = \frac{k}{2\Delta y^2}\mathbf{D}$, $\alpha_z = \frac{k}{2\Delta z^2}\mathbf{D}$, Δx, Δy and Δz are the grid spacings in the x, y and z directions, respectively, and

$$\begin{aligned}\mathbf{T}^n_{i,j,k} &= 2\alpha_x(\mathbf{U}_{i+1,j,k} - 2\mathbf{U}_{i,j,k} + \mathbf{U}_{i-1,j,k}) \\ &+2\alpha_y(\mathbf{U}_{i,j+1,k} - 2\mathbf{U}_{i,j,k} + \mathbf{U}_{i,j-1,k}) \\ &+2\alpha_z(\mathbf{U}_{i,j,k+1} - 2\mathbf{U}_{i,j,k} + \mathbf{U}_{i,j,k-1}) + \mathbf{S}^n_{i,j,k}\end{aligned} \tag{5}$$

Since the finite difference discretization of a system of $N_e(= 4)$ partial differential equations results in $N_e \times N_x \times N_y \times N_z$ nodal variables and algebraic equations. A main issue when solving systems of linear algebraic equations such as Eqs. 4 is the ordering of the equations and variables. A natural ordering of the grid points and blocking of nodal variants has been chosen here to obtain a well structured matrix and because the cache behavior is better. For this ordering, the system of equations 4 can be expressed in matrix form as $Ax = b$, where A is an heptadiagonal block matrix, with 4×4 blocks, because $N_e = 4$ in Eq 1, and $x = \Delta\mathbf{U}$. For illustration, Fig. 1 shows the pattern of the matrix A for an equally-spaced meshes of size $5 \times 5 \times 5$.

We have used the Generalized Minimal Residual (GMRES) method [11] for solving the linear system of equations, since it is an appropriate method for non-symmetric linear systems. In addition, to ensure its convergence in few iterations we have used the Jacobi method as preconditioner.

2.1 Simulation Results

Computations have been performed on $51 \times 51 \times 51$ and $101 \times 101 \times 101$–point equally-spaced meshes. The choice of these grids is based on that $51 \times 51 \times 51$ is the minimum resolution to achieve sufficiently accurate results, and the

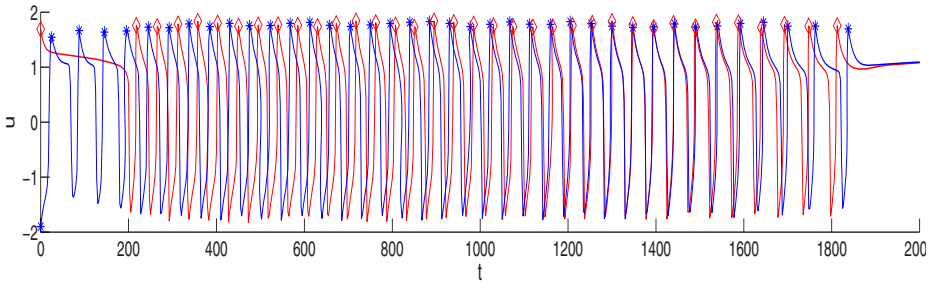


Fig. 2. u versus the number of time steps t in ω ($*$) and $\Omega - \omega$ ($\diamond$)

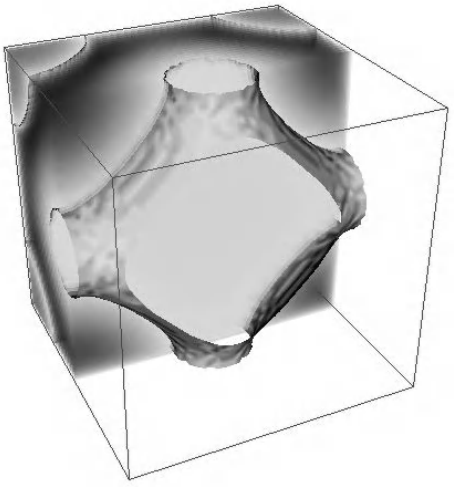


Fig. 3. Three-dimensional visualization of u

$101 \times 101 \times 101$ mesh has been considered to extend the evaluation for grids of larger size. Here after, we will refer to these equally-spaced meshes by coarse grid and fine grid respectively. The time step is equal to 0.0001 s, in order to assess the spatial accuracy. Some sample results are presented in Figs. 2 and 3. Fig. 2 shows the time history of u at two monitor points: one (identified with asterisks) in ω and the other (identified with squares) in $\Omega - \omega$. This figure shows that, for the initial conditions considered in this paper, the time history of u in $\Omega - \omega$ exhibits an oscillatory behavior characterized by periods of intensive variations in u from $t = 0$, whereas the time history of u in ω takes a while before it starts oscillating. The oscillations observed in Fig. 2 are typical of relaxation oscillations.

Fig. 2 also shows that the oscillations of u in ω and $\Omega - \omega$ are not in phase, but nearly coincide at about $t = 1370s$; thereafter, the one that was lagging overcomes the other, and, after some time, both time histories reach the same

slowly varying value. Similar trends to those shown in Fig. 2 have also been observed at other six monitoring locations.

Although not shown here, three–dimensional, time–dependent visualizations of u indicate that there are inward and outward propagating waves in ω before the outward ones propagate in $\Omega - \omega$, i.e., there is an interplay between the values of u in ω and $\Omega - \omega$ caused by the diffusion of the dependent variables. This back–and–forth play between the inner and outer domains is a consequence of the initial conditions, the source/reaction terms, and the diffusion coefficients of Eqs. 1. A snapshot illustrating u throughout the domain is presented in Fig. 3 which shows steep complex patterns.

3 Parallel Implementation

Our parallel formulation relies on using domain decomposition techniques, whereby the computational domain is divided into as many domains as the number of processors. We have adopted the message passing paradigm based on MPI, since it still provides the best performance over a large range of applications on several parallel platforms [10]. Specifically, the proposed parallel algorithm consists of the following steps:

Each processor computes its initial local subdomain.
do n=1,k
- Each processor updates its corresponding submatrix and right hand side.
- Solving the system of equations in parallel using the preconditioned GMRES, included in PETSc.
- Each processor communicates to its adjacent ones its local boundary, by point to point communications.

end do

We have used the GMRES solver and Jacobi preconditioner, both included in the free portable extensible toolkit for scientific computation (PETSc) library [9]. PETSc software is a freely available tool that integrates a hierarchy of components, including data structures, Krylov solvers and preconditioners that can be embedded into various code. This work has not been limited to the integration of PETSc routines into the code but has also adapted it to the characteristics of the problem to optimize their performance.

Before carrying out the parallel implementation we have profiled the serial code to identify the most computationally-consuming step. The profiling results have shown that the solution of the system of equations is computationally the most predominant step, requiring about 80% of the computational time of each single time step. Clearly, an efficient parallel implementation of the code will depend principally on the efficiency of the solution of the system of equations. Additionally, the choice of an appropriate solving method, as well as, the sparse matrix storage format can improve drastically the efficiency. In this work, the parallel sparse matrix, A, was stored in Compressed Sparse Row (CSR) format and the linear system of equations was solved using the GMRES method preconditioned by Jacobi method.

4 Evaluation of the Parallel Implementation

We have assessed the performance of the parallel three–dimensional bursting simulation for both coarse and fine grids on the architectures described below:

(a) A cluster of five Intel(R) Xeon(TM) biprocessors at 3.06 GHz with 2 GB RAM and 512 KB cache. The nodes of the cluster are interconnected via two gigabit Ethernet networks, one for data (NFS) and the other for computation. The architecture of this cluster is based on a UMA access, where all processors have equally fast (symmetric) access to the memory in the node.
(b) A SGI Altix 3700 Bx2 of 8 processors at 1600 MHz Intel Itanium 2 Rev with 128 GB RAM, 32KB L1, 256KB L2 and 6.00MB L3. The Altix 3700 computer system is based on a Distributed Shared Memory architecture [7] and uses a cache–coherent Non-Uniform Memory Access (NUMA) where the latency of processors to access the local memory is lower than the latency to access the global memory (or remote memory) [8].

Preliminary evaluation has shown that the computational cost is similar for each time step, therefore we have fixed the number of time steps in five to expedite the evaluation process.

Table 1. Performance and speedup of the parallel simulation on (a) cluster of 5-biprocessors SMP and (b) 8-processors SGI Altix 3700 Bx2 respectively, for the coarse and fine grids

	(a) Cluster of 5-biprocessors				(b) SDM Platform			
	coarse grid		fine grid		coarse grid		fine grid	
proc	runtime (s)	speedup	runtime (s)	speedup	runtime (s)	speedup	runtime (s)	speedup
1	2.70	1.00	22.63	1.00	2.57	1.00	21.68	1.00
2	2.00	1.34	12.46	1.81	1.47	1.74	13.76	1.57
4	1.07	2.50	6.70	3.37	0.81	3.17	6.86	3.15
6	0.84	3.19	4.76	4.74	0.53	4.79	4.48	4.82
8	0.69	3.90	3.91	5.78	0.47	5.42	3.60	6.01
10	0.59	4.56	3.35	6.73				

On the cluster of 5-biprocessors SMP, the parallel performance and the scalability are better when the dimension of the matrix is larger, as shown in Table 1(a). One of the underlying reasons is the increasing ratio computation versus communication: fine grid (i.e. large matrix) involves more computation, but also a high increment of communications, see Table 2.

On the 8-processor SGI Altix 3700 Bx2, see Table 1(b), generally, the performance achieved by the parallel simulation, for both grids is good. Besides, the coarse grid exhibits slightly better scalability for less than four processors, this fact can be explained by the good behavior of the memory management,

Table 2. Percentage of communications for both coarse and fine grids, on a cluster of 5-biprocessors

proc	% of communications	
	coarse grid	fine grid
2	25 %	1 %
4	24 %	6 %
6	30 %	19 %
8	42 %	15 %
10	33 %	26 %

since the percentage of data that fits in cache is higher compared to the fine grid. Moreover, the scalability is better than on the SMP cluster, one of the underlying reasons is the high speed interconnection technology used in this architecture [8].

The performance of our simulation on the platform based on Itanuim is less efficient than expected, since the performance results showed by the Standard Performance Evaluation Corporation (spec) [12] for several benchmarks on Itanuim are over a factor of two better than on systems based on Xeon. This low efficiency can be explained by the bad behavior of the CSR storage on Itanuim for matrices of short rows [13].

5 Conclusions and Future Works

In this work, we have presented a mathematical model and its parallel implementation to simulate a three–dimensional bursting phenomena. This model is described by four, nonlinearly coupled partial differential equations. The differential equations have been discretized by means of a linearly–implicit finite difference method in equally–spaced grids. The resulting system of linear algebraic equations at each time level has been solved by means of the GMRES solver with Jacobi preconditioning, both included in PETSc. By the implementation of the proposed model, we have determined the homogeneous initial conditions that exhibits bursting and nonhomogeneous ones that exhibit both bursting and quenching.

The passing message paradigm has been adopted to carry out the parallel implementation on two parallel platform (1)a DSM platform, based on cc-NUMA memory access, and on (2) a cluster of SMPs based on UMA access. From the evaluation of the parallel simulation, we conclude that the code achieves better scalability on the DSM platform. This conclusion can be translated to SPMD applications that are based on the solution of PDE for similar dimensions.

Currently, we are working on the use of new storage format, specifically, we use the Compressed Diagonal Storage, to improve the performance of the solution of the sparse system of equations on Systems based on Itanuim.

References

1. Rinzel, J., Lee, Y.S.: Dissection of a model for neuronal parabolic bursting. Journal of Mathematical Biology 25, 653–675 (1987)
2. Rinzel, J.: Electrical excitability of cells, theory and experiment: review of the Hodgkin–Huxley foundation and an update. Bulletin of Mathematical Biology 52, 5–23 (1990)
3. Smolen, P., Keizer, J.: Slow voltage inactivation of Ca^+2 currents and bursting mechanisms for the mouse pancreati β–cell. Journal of Membrane Biology 127, 9–19 (1992)
4. Bertram, R., Buttle, M.J., Kiemel, T., Sherman, A.: Topological and phenomenological classification of bursting oscillations. Bulletin of Mathematical Biology 57, 413–439 (1995)
5. Keener, J., Sneyd, J.: Mathematical physiology. Springer, Heidelberg (1998)
6. Ramos, J.I.: Linearization methods for reaction–diffusion equations: Multidimensional problems. Appl Math Comput 88, 225–254 (1997)
7. Protic, J., Tomasevi, M., Milutinovic, V.: Distributed Shared Memory: Concepts and Systems. Wiley, Chichester (1997)
8. Dunigan, T., Vetter, J., Worley, P.: Performance evaluation of the SGI Altix 3700. In: Proceedings of the IEEE Intl. Conf. Parallel Processing, ICPP, pp. 231–240 (2005)
9. Balay, S., Buschelman, K., Gropp, W., Kaushik, D., Knepley, M., McInnes, L.C., Smith, B., Zhang, H.: PETSc Users Manual.Revision 2.1.5. Argonne National Laboratory, ANL-95/11 (2002)
10. Krawezik, G., Cappello, F.: Performance comparison of MPI and OpenMP on shared memory multiprocessors. Concurrency and Computation: Practice and Experience 18, 29–61 (2006)
11. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM (2003)
12. `//www.spec.org/`
13. Mellor-Crummey, J., Garvin, J.: In Optimizing Sparse Matrix-vector Product Computations Using Unroll and Jam. International Journal on High Performance Computing Applications 18(2) (2004)

PAR Reduction Scheme for Efficient Detection of Side Information in OFDM-BLAST System

Myung-Sun Baek[1], Sang-Tea Kim[2], Young-Hwan You[1], and Hyoung-Kyu Song[1]

[1] uT Communication Research Institute, Sejong University, Seoul, Korea
[2] Korea Institute of Industrial Technology Evaluation & Planing
sabman@sju.ac.kr, kst@itep.re.kr, yhyou@sejong.ac.kr, songhk@sejong.ac.kr

Abstract. In this paper, we investigate the PAR reduction scheme for efficient detection of side information in OFDM-based layered space-time system using the SLM approach. The investigated SLM-based OFDM-BLAST system selects the transmitted sequence with the lowest average PAR over all transmitting antennas and retrieves the side information very accurately at the expense of a slight degradation of the PAR performance. The low probability of false side information can improve the overall detection performance of the OFDM-BLAST system with erroneous side information compared to the conventional SLM approach.

1 Introduction

The goal of future wireless communication is to provide high quality wireless multimedia services, and thus high data rate communication must be transmitted reliably. Orthogonal frequency division multiplexing (OFDM) is commonly used for high data rate wireless communication due to its inherent error susceptibility in a multipath environment and has been chosen for several broadband WLAN standards like IEEE802.11a, European HIPERLAN/2, and Japanese MMAC, and terrestrial digital audio broadcasting (DAB) and digital video broadcasting (DVB-T) [1]-[3]. A combination of multiple input multiple output (MIMO) signal processing with OFDM is regarded as a promising solution for enhancing the performance of next generation wireless communications [6]-[8]. However, the high peak-to-average power ratio (PAR) in orthogonal frequency division multiplexing (OFDM) modulation systems can significantly reduce power efficiency and performance. To reduce the PAR of OFDM system, several techniques such as selected mapping (SLM) and partial transmit sequences (PTS), etc., have been proposed. The SLM improves the PAR statistics by introducing a little redundancy and also solves the problem of signal distortion techniques[2]. In the SLM-based multiple input multiple output (MIMO) OFDM systems, since two or more OFDM signals are transmitted simultaneously, it is important to reduce complexity of PAR reduction scheme and to detect side information (SI) of multiple transmitting antennas accurately.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 950–957, 2007.
© Springer-Verlag Berlin Heidelberg 2007

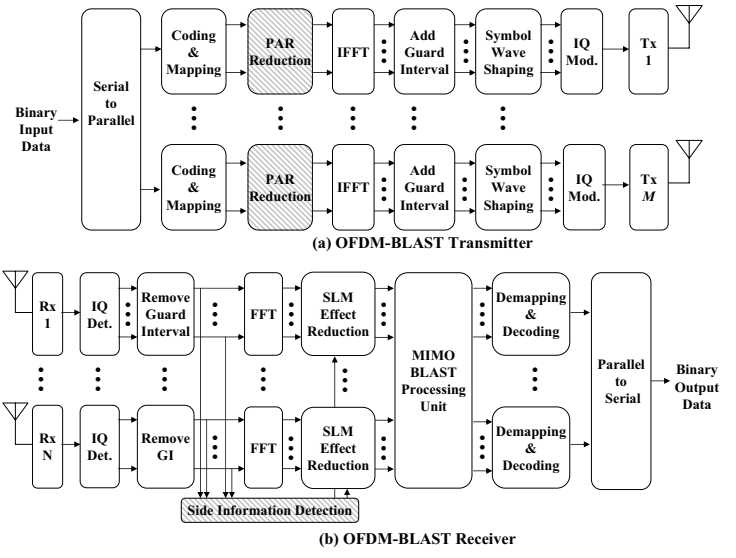


Fig. 1. OFDM-BLAST system model with SLM PAR reduction

In this paper, we address the PAR performance of the OFDM-based Bell laboratories layered space-time (BLAST) system using the SLM scheme for PAR reduction. The conventional method is to apply the SLM scheme for each transmitting antenna individually. We investigate the modified SLM (MSLM) approach in the OFDM-BLAST system, which selects the transmitted sequence with the lowest average PAR over all transmitting antennas. The OFDM-BLAST system based on MSLM recovers the SI with the low error probability at the expense of a slight degradation of the PAR performance and the overall bit error rate (BER) performance in the presence of erroneous SI can be improved by exploiting the receiver diversity.

2 PAR Definition in MIMO-OFDM Systems

As illustrated in Fig. 1, we consider an OFDM-vertical BLAST (VBLAST) system with K subcarriers that uses M transmitting and N receiving antennas [1]. If we denote $X_i(k)$ for the frequency-domain index $0 \leq k < K$ as a data pattern for the i-th transmitting antenna for $1 \leq i \leq M$, $x_i(k)$ is the inverse fast Fourier transform (IFFT) output of the data pattern for the i-th transmitting antenna. The relationship is derived as following rule:

$$x_i(k) = \mathrm{IFFT}[X_i(k)] \tag{1}$$

where $\mathbf{x}_i = [x_i(0), x_i(1), \cdots, x_i(k), \cdots, x_i(K-1)]^T$. Peak power is generally evaluated in terms of PAR, which is given by [3]

$$\mathrm{PAR} = \frac{\max\limits_{0\le k\le K} |x_i(k)|^2}{E[|x_i(k)|^2]}, \tag{2}$$

where $E[\cdot]$ denotes the expected value. Applying the central limit theorem, the time-domain samples $x_i(k)$ with large values of K become Gaussian distributed. Assuming that the samples are mutually uncorrelated, the probability that the PAR exceeds a certain threshold PAR_0 is defined as [2]

$$\Pr[\mathrm{PAR} > \mathrm{PAR}_0] = 1 - \left(1 - e^{-\mathrm{PAR}_0}\right)^K. \tag{3}$$

3 Ordinary SLM

In the ordinary SLM, we assume that G statistically independent OFDM symbol represents the same information. In the ordinary SLM-based MIMO-OFDM system, the SLM approach can be applied to each transmitting antenna independently. The core of selected mapping is to choose one particular signal that exhibits some desired properties out of K signals representing the same information. After mapping the information to the subcarrier $X_i(k)$, each OFDM symbol is multiplied with the G vectors $\mathbf{V}^{(g)} = [V_0^{(g)}, \ \cdots, \ V_{K-1}^{(g)}]^T$ with each component $V_k^{(g)} \in \{\pm 1, \pm j\}$ for $1 \le g \le G$, resulting in a set of $M \cdot V$ different OFDM symbols $\mathbf{X}_i^{(g)}$ represented by

$$\mathbf{X}_i^{(g)} = \mathbf{X}_i \cdot \mathbf{V}^{(g)}, \ 1 \le g \le G. \tag{4}$$

All new OFDM symbols are transformed into time domain using IFFT defined by $x_i^{(g)}(k) = \mathrm{IFFT}[X_i^{(g)}(k)]$ and then symbol $\widetilde{\mathbf{x}}_i$ with the lowest PAR is selected.

The G independent OFDM symbols $\mathbf{x}_i^{(g)}$ for i-th transmitting antenna represent the same information and the symbol $\widetilde{\mathbf{x}}_i$ with the lowest PAR is selected for transmission to meet the following condition

$$\widetilde{g}_i = \underset{1\le g\le G}{\mathbf{argmin}} \left[\max_{0\le k<K} |x_i^{(g)}(k)|\right] \tag{5}$$

where $\widetilde{g}_i$ is an index of symbol with the lowest PAR among G independent OFDM symbols $\mathbf{x}_i^{(g)}$ for i-th transmitting antenna. The transmitted symbol $\widetilde{\mathbf{x}}_i$ is equal to $\mathbf{x}_i^{(g_i)}$ with the lowest PAR. Selecting the symbol with the lowest PAR for transmission, the probability that $\mathrm{PAR}_{\mathrm{low}}$ exceeds PAR_0 is given by

$$\Pr[\mathrm{PAR}_{\mathrm{low}} > \mathrm{PAR}_0] = \left[1 - \left(1 - e^{-\mathrm{PAR}_0}\right)^K\right]^G. \tag{6}$$

In order to demodulate received data, the receiver has to know information about $\mathbf{V}^{(g_i)}$ of M transmitting antennas and transmitter therefore has to transfer the SI of each $\widetilde{\mathbf{g}}_i$. The number of required bit for sending $\widetilde{\mathbf{g}}_i$ per each transmitting antenna is $\log_2 G$ bits [11]. Corresponding SI bits in the i-th transmitting antenna are inserted in each OFDM frame $\mathbf{X}_i$ and then are transmitted.

4 Simplified SLM

In the simplified SLM-based MIMO-OFDM system, the M subcarrier vectors $\mathbf{X}_i$ from M transmitting antennas are multiplied subcarrierwise with the same vector, which is one of the G vectors $\mathbf{V}^{(g)} = [V^{(g)}(0), \ \cdots \ V^{(g)}(K-1)]^T$ for $1 \leq g \leq G$, resulting in G sets composed of M different subcarrier vectors. For each set, the M different subcarrier vectors are transformed by M separate K-points IFFTs and a set with the lowest average PAR overall M transmitting antennas is chosen as follows

$$\widetilde{g} = \underset{1 \leq g \leq G}{\mathbf{argmin}} \left(1/M \sum_{i=1}^{M} \max_{0 \leq k < K} \left| x_i^{(g)}(k) \right| \right) \tag{7}$$

where $\widetilde{g}$ is an index of symbol with the lowest PAR overall M antennas. Selecting the set with the lowest average PAR for transmission, the probability that the lowest average PAR of overall M transmitting antennas exceeds PAR_0 is given by

$$\Pr[\mathrm{PAR}_{\mathrm{low}} > \mathrm{PAR}_0] = \left[1 - \left(1 - e^{-\mathrm{PAR}_0} \right)^{M \cdot K} \right]^G. \tag{8}$$

In order to demodulate received data, the receiver has to know information about $\mathbf{V}^{(g)}$ and transmitter therefore has to transfer the SI $\widetilde{g}$. Because the M subcarrier vectors with the lowest average PAR are constructed by multiplying the corresponding information sequence of each transmitting antenna by the same random vector, it is possible that only one transmitting antenna bears the SI bits. Therefore the receiver diversity effect can be applied by receiving the same N times information bits per each receiving antenna [12].

5 Detection of Side Information

At the receiving side, in order to recover the transmitted data, the receiver has to know the generation of the transmitted OFDM signal from the embedded SI. In the MSLM-based OFDM-BLAST system, one transmitting antenna can bear the SI bits and N receiving antennas receive the same information bits. Therefore, by using the maximum ratio combining (MRC) receiver diversity, the probability of the erroneous detection for the SI bits can be obtained as follows

$$P_{f,\ \mathrm{SLM}} \approx \log_2 G \cdot P_b \tag{9}$$

with

$$P_b = [1/2(1-\mu)]^N \cdot \sum_{\omega=0}^{N-1} \binom{N-1+\omega}{\omega} [1/2(1+\mu)]^\omega \tag{10}$$

where $\mu = [\gamma_0/(1+\gamma_0)]^{-1/2}$, and γ_0 is the average signal-to-noise ratio (SNR) per antenna [12], [13]. In MSLM-based OFDM-BLAST system with over two receiving antennas ($N \geq 2$) a receiver diversity effect is obtained by receiving the same SI bits in the N receiving antennas.

If we assume that the G vectors $\mathbf{V}^{(g)}$ for $1 \leq g \leq G$ are statistically independent in the SLM case and the side information is false, the conditional BER of the OFDM-BLAST system is approximately 1/2. Then, an overall BER, denoted by $P_{e,\mathrm{SLM}}$, can be approximated by

$$P_{e,\ \mathrm{SLM}} \approx P_e \cdot (1 - P_{f,\ \mathrm{SLM}}) + 1/2 P_{f,\ \mathrm{SLM}} \tag{11}$$

where P_e is BER of the OFDM-BLAST system with M transmitting antennas that the SI is correctly detected. In eqn. (11), the false detection probability of SI bits is very low due to the MRC diversity. Therefore, the overall BER performance significantly depends on the error probability P_e.

6 Examples and Discussions

In this section, we evaluate the PAR performance and the probability of erroneous detection for the SI in OFDM-BLAST system using the SLM approach. The simulations were performed with 256 subcarriers for QPSK modulation.

Fig. 2 shows the probabilities which the PAR exceeds a certain threshold PAR_0 in the conventional and modified SLM approaches. For $\Pr[\mathrm{PAR}_{\mathrm{low}} > \mathrm{PAPR}_0] = 10^{-3}$, in the case of $M = 4$, the PAR performance of the MSLM approach with $G = 2$ and 4 is degraded by 0.6dB and 1.1dB compared to that of the conventional SLM approach with $G = 2$ and 4, respectively. As the number of the transmitting antennas is increased, as expected, the PAR performance is decreased regardless of G.

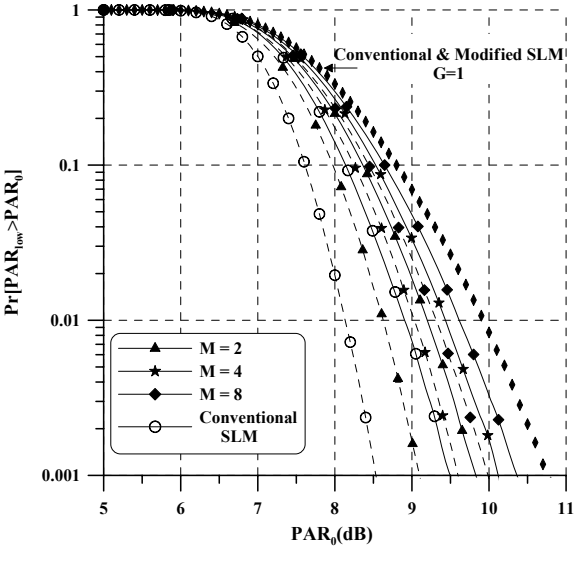


Fig. 2. PAR performance of OFDM-BLAST systems using both ordinary and modified SLM approaches according to the number of antennas (1) Solid lines - $G = 2$, (2) Dash lines - $G = 4$

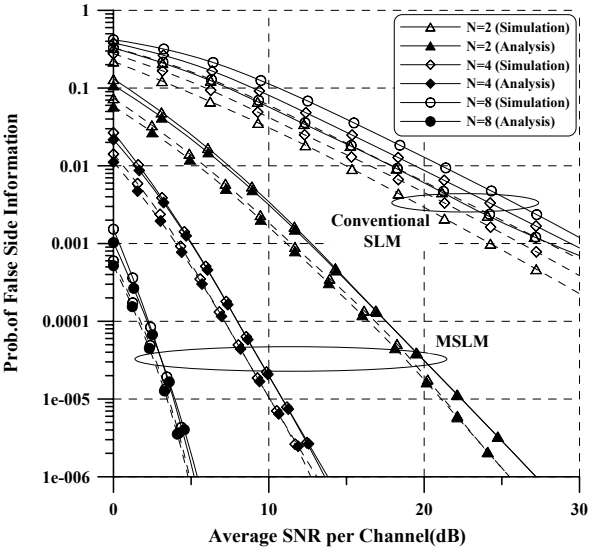


Fig. 3. Probability of the erroneous detection of the side information for both conventional and modified SLM approaches : (1) Solid lines - $G = 4$, (2) Dash lines - $G = 2$

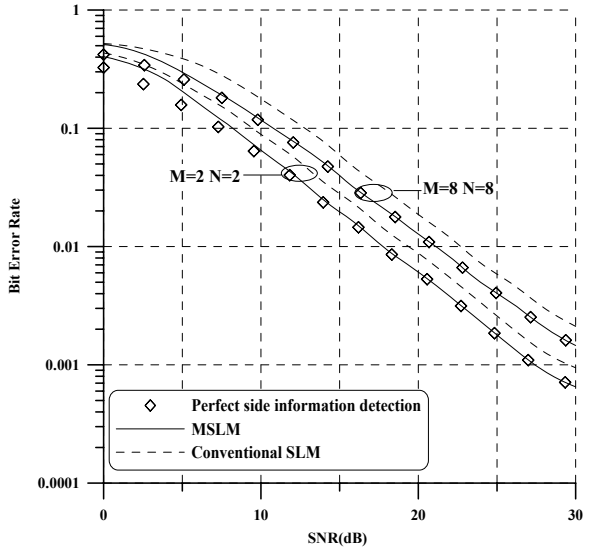


Fig. 4. Overall BER performance of the OFDM-BLAST system with respect to side information error probability for both conventional and modified SLM approaches according to the number of antennas with G = 4

The probability of false SI for both conventional and modified SLM is illustrated in Fig. 3. It can be also seen that the analytic error curves according to eqn. (11) are very close to the simulated error curves. Furthermore, the detection probability of SI bits in the MSLM can be dramatically improved due to the receiver diversity effect according to the increase of receiving antennas, compared to that of the conventional SLM. Also, the performance degradation is found according to the increase of G, regardless of PAR reduction schemes and the number of transmitting antennas.

Fig. 4 shows the overall BER performance due to the erroneous SI for both conventional and modified SLM. In the case of SLM approach with $G = 4$, for $E_b/N_0 > 10dB$, the overall BER performance of the simplified SLM-based OFDM-BLAST system is similar to that of perfect SI detections. On the other hand, the conventional SLM-based OFDM-BLAST system experiences the overall SNR loss. The improvement of MSLM-based OFDM-BLAST system may be introduced by using the MRC receiver diversity detecting the SI bits.

7 Conclusions

In this paper, the improved PAR statistics can be obtained by approaching the SLM scheme, which are less complex compared to the ordinary approache. From the simulation results, we have shown that the MIMO-OFDM system with investigated PAR reduction scheme decodes the SI very accurately. The presented results are valid for other OFDM systems like some broadband wireless local area network (LAN) standards like IEEE802.11a and IEEE802.11g, and terrestrial digital audio broadcasting (DAB) and digital video broadcasting (DVB-T).

Acknowledgement

This research is supported by the Ubiquitous Autonomic Computing and Network (UCN) Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea

References

1. van Nee, R., Awater, G., Morikura, M., Takanashi, H., Webster, M., Halford, K.W.: New high-rate wireless LAN standards. IEEE Commun. Mag. , 82–88 (1999)
2. Rado Broadcasting Systems: Digital Audio Broadcasting (DAB) to Mobile, Portable and Filxed Receivers, ETSI, ETSI ETS 300401 (1995)
3. Digital Video Broadcasting (DVB): Frame Structure, Channel Coding and Modulation for Digital Terrestrial Television (DVB-T), ETSI, ETSI ETS 300 744 (1997)
4. Alamouti, S.M.: A simple transmit diversity technique for wireless communications. IEEE 16(8), 1451–1458 (1998)
5. Foschini, G.J.: Layered space-time architecture for wireless communications in a fading environment when using multi-element antennas. Bell Labs Technical Journal 1(2), 41–59 (1996)

6. Larsson, E.G., Li, J.: Preamble design for multiple-antennas OFDM-based WLANs with null subcarriers. IEEE Signal Processing Letters 8(11), 285–288 (2001)
7. Li, Y(G.): Simplified channel estimation for OFDM systems with multiple transmit antennas. IEEE Trans. Wireless Communications 1(1), 67–75 (2002)
8. Piechocki, R.J., Fletcher, P.N., Nix, A.R., Canagarajah, C.N., McGeehan, J.P.: Performance evaluation of BLAST-OFDM enhanced Hiperlan/2 using simulated and measured channel data. Electronics Letters 37(18), 1137–1139 (2001)
9. Bauml, R.W., Fischer, R.F.H., Huber, J.B.: Reducing the peak-to-average power ratio of multicarrier modulation by selected mapping. Electron. Lett. 32(22), 1257–2056 (1996)
10. Kwon, O.-J., Ha, Y.-H.: Multi-carrier PAP reduction method using sub-optimal PTS with threshold. IEEE Trans. Broadcasting 44(2), 232–236 (2003)
11. Pushkarev, P.A., Ryu, K.-W., Yoo, K.-Y., Park, Y.-W.: A study on the PAR reduction by hybrid algorithm based on the PTS and SLM techniques. In: IEEE Vehicular Technology Conference, 2003. vol. 2, pp. 1263–1267 (2003)
12. Kim, C.-J., Kim, Y.-S., Jung, C.-Y.: BER analysis of QAM with MRC space diversity in Rayleigh fading channel. In: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, vol. 2, pp. 482–485 (1995)
13. Proakis, J.G.: Digital Communications. McGraw-Hill Inc., New York (2001)

Fuzzy PI Controller for Turbojet Engine of Unmanned Aircraft

Min Seok Jie, Eun Jong Mo, and Kang Woong Lee

School of Electronics, Telecommunication and Computer Engineering, Hankuk Aviation University, 200-1, Hwajeon-dong, Deokyang-gu, Koyang-city, Kyonggi-do, 412-791, Korea
Tel.: +82 2 3158 0166, Fax : +82 2 3159 9257
tomsey@korea.com, moe@pweh.com, kwlee@mail.hangkong.ac.kr

Abstract. This paper proposes a turbojet engine controller for unmanned aircraft using Fuzzy-PI(Proportional Integral) algorithm. To prevent surge or flame out which could occur during engine acceleration or deceleration operation, the PI type fuzzy system controls the fuel flow input value. The fuzzy logic with logarithm function is designed to obtain the fast acceleration and deceleration of the engine under the condition that the operating point should stay between the surge and flame out control lines during the transient. This paper presents a fuzzy inference rule made by the logarithm function can improve the characteristics of tracking error in steady state. The proposed control scheme is proved by computer simulation using a linear engine model. Simulation results on the state space model of a turbo jet engine illustrate that the proposed controller achieves the desired performance.

Keywords: fuzzy control, logarithm function, turbojet engine, unmanned aircraft, fuel flow control.

1 Introduction

It is obvious that different control approach of aircraft engine is applied depending upon the purpose of the flight or flight condition such as starting, take-off, or landing operation. Depending on the external factors such as air temperature or pressure, the engine control scheme may be different as well.

For military aircraft, sharp maneuvering is prerequisite requirement of flight control, but for the commercial aircraft, smooth maneuvering is required. The application of different control approaches has been widely studied to achieve flight safety and reliability.

The method to control aircraft engine in the past was very limited to basic operations such as engine start or thrust control. With the introduction of automatic engine control system as well as high performance applications, the development of engine control technology is becoming apparent. For applying automatic engine control system into both military and commercial aircraft engine, it is required to quick response characteristics to any change of speed or thrust during the engine acceleration or deceleration. Automatic control of fuel flow or compressor rotating speed is applied to prevent surge which is occurred in high thrust operation

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 958–967, 2007.
© Springer-Verlag Berlin Heidelberg 2007

environment such as take off or landing. In addition, the automatic engine control system controls rotor speed by measuring the inlet and outlet temperature. As a result of that, it minimizes the distress of the parent material of the engine so that it can achieve engine life cycle extension as well as fuel efficiency.

As mentioned above, to automatically control an aircraft engine, each system should be realized into digital format. Fuel flow control is proposed to control rotor speed automatically. The desired thrust can be achieved by varying the fuel flow or exhaust nozzle dimension, which is recently proved by applying multivariable control approach [1]. Application of multivariable control approach is used to design for linear models using LQG/LTR [2], [3] approach or H-infinity control [4] synthesis formulation. This technique is a robust control to reduce disturbance or parameter variations, however, it may weaken control performance because of uncertainty of each engine model which is linearized at each operating point.

However, small engine such as unmanned aerial vehicle is required to system simplicity by limiting the dimension of exhaust nozzle so that the engine control system suited to SISO(Single Input Single Output) system [5] can be designed to achieve desired engine performance by controlling fuel flow effectively.

One important factor to be considered to design engine control system is to design a protection system which controls the engine not to occur surge or flame out during engine acceleration or deceleration. When high thrust is applied, like take-off operation, surge can happen and it is one of the major causes to lead flight accident or incident. Therefore it is important to consider implementing a surge prevention system in the aircraft engine. To prevent surge effectively, engine acceleration should be controlled to run under the surge control line. In this case, if the reference command is set as a compressor rotating speed in the initial stage, the engine may run exceed the surge margin by sudden acceleration, however through surge control system, the fuel flow is controlled to reduce the engine rotating speed to drop it under the surge limit.

This paper presents a design approach applying fuzzy inference method to a small turbo jet engine for unmanned aircraft in which fuel flow is used as only control input. To prevent surge which would be occurred during engine acceleration, this system includes 1) setting a reference acceleration speed for surge prevention, and then 2) improving error characteristics in steady state by log scale of residual errors between the reference acceleration speed and the actual compressor rotating speed, and 3) deciding fuel flow control input using fuzzy inference method which control the actual acceleration speed follow the reference acceleration speed. The proposed control capabilities are proved by simulation using linear engine model.

2 Turbo Jet Engine System and Fuzzy Inference Control

2.1 Turbo Jet Engine System

Simplified turbo jet engine consists of the three main stages as shown in Fig. 1. In the model considered, the air is compressed by compressors. The compressed air then enters a combustion chamber where it undergoes combustion with injected fuel. The subsequent decompression takes place in turbines, where part of the thermodynamic

energy of the gas is transferred through the engine shaft back to other mechanical parts. An important energy balance is that the energy needed for air compression comes from the energy transferred to the shafts in the turbines. If the engine is in steady-state, then the energy needed in the compressors is equal to the energy produced in the turbines. Since the engine thrust is in proportion to the rotor speed of compressor, the engine thrust can be estimated by measuring the rotor speed of compressor and the estimated thrust value can be applied to engine control system.

The engine acceleration and deceleration control for generating the desired thrust is designed to control fuel flow so that the desired thrust can be controlled under the command of speed which is transferred into the rotor speed of compressor.

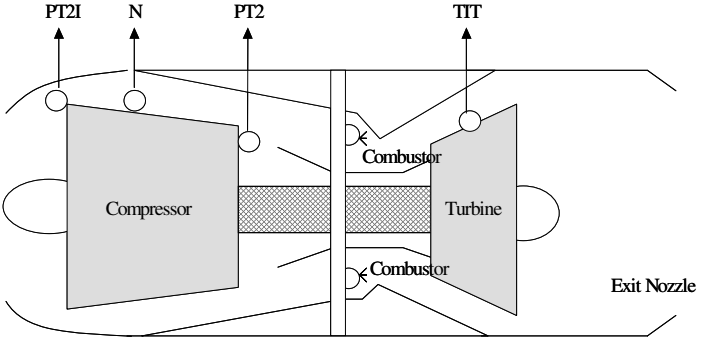


Fig. 1. Schematic of a turbojet engine

2.2 Fuzzy Inference Control

During engine acceleration or starting, pressure of the compressor is decreased by increasing of air mass. As a result of this, air may turn back and flow from the smooth to turbulent. This disturbed state in which the energy supplied to the compressor is simply wasted in churning the air around is known as surging. To control surge effectively, the engine acceleration should be controlled to follow the surge control line in which the surge margin line is closely positioned.

As shown in Fig. 2., acceleration control is achieved by setting a surge margin line in which sufficient surge margin is secured, and having engine control line to be positioned near the surge control line. In this case, if the reference command is set up as a compressor rotating speed, then in the initial stage the engine may run over the surge margin by sudden acceleration, however through the surge control, the fuel flow is controlled. As a result of that maintaining of slow engine acceleration is achieved.

For surge control, it is required to establish surge limit about 90% of the surge margin and select an engine acceleration speed as a reference command which controls the engine operate under the operation line not to run over the surge margin line, and select a flue flow control system which controls the compressor acceleration speed to control the compressor rotating speed follow the reference command.

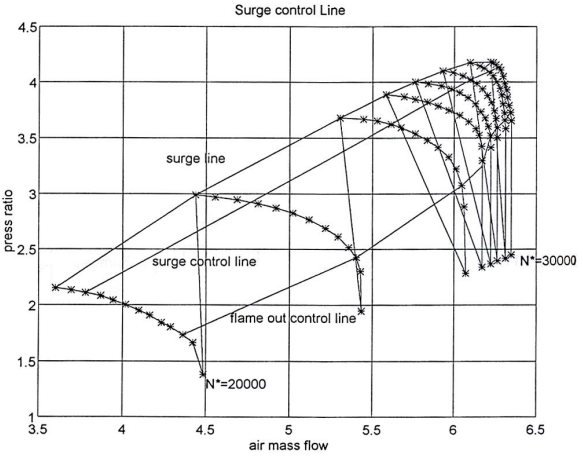


Fig. 2. Surge control line

If the reference speed is used as a control command, the surge may be easily occurred by sudden engine acceleration. Therefore, it would be preferable to use the reference acceleration as a control command, so that the engine control is achieved around the surge control line.

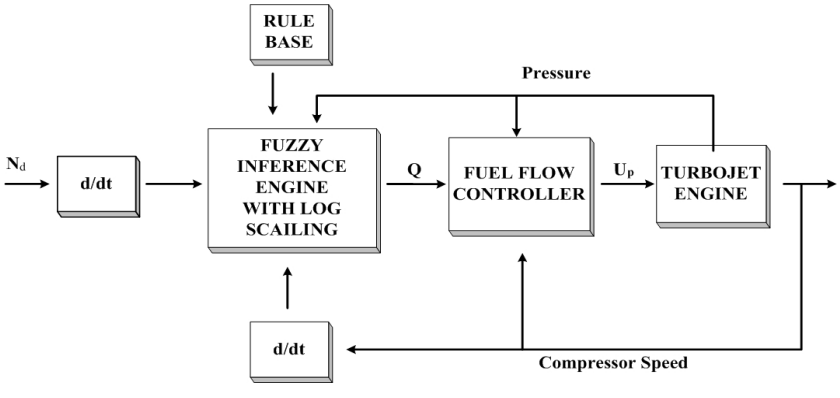


Fig. 3. Structure of the engine control system using fuzzy-inference method

Fig. 3., shows the structure of the engine control system using fuzzy inference method where $\dot{N}_d$ is reference acceleration on which the surge margin is reflected. We define $e_N = \log_{10}(\dot{N}_d - \dot{N})$ as the error of reference acceleration speed to the actual acceleration speed. We present the method to improve steady state error of compressor rotating speed by log scaling the error of compressor rotating acceleration speed.

And, the fuzzy inference fuel flow control input is described using PI control structure as follows;

$$u_p(t) = K_p(e_N, e_p)e_N + K_I(e_N, e_p)\int_0^t e_N(\tau)d\tau \tag{1}$$

where, e_p is an error of compression ratio at inlet of compressor. The control gain K_p and K_I are chosen as fuzzy inference method using e_N, e_p.

Fuzzy inference input variable defines e_N, e_p, the output variable defines PI control gain weight value Q.

The Table 1. defines the fuzzy rule of the input fuzzy variables and output fuzzy variables, and divides language value of input variable into 7 steps and the output fuzzy variables into 5 steps and the membership function is defined as Fig. 4.

The output fuzzy variables obtained from applying fuzzy rule are used to control input through de-fuzzification.

Table 1. Fuzzy rules

e_P \ e_N	NB	NM	NS	ZE	PS	PM	PB
NB	VB	VB	VB	VB	VB	VB	VB
NM	VB	VB	B	M	B	VB	VB
NS	VB	B	M	S	M	B	VB
ZE	B	M	S	VS	S	M	B
PS	VB	B	M	S	M	B	VB
PM	VB	VB	B	M	B	VB	VB
PB	VB	VB	VB	VB	VB	VB	VB

If the language values of fuzzy input variables are $e_N = x_1^0, e_p = x_2^0$ and the comparability of the condition "x is A" is $A(x^0)$, then the compatibility of the two inputs are defined using Mamdani method.

$$W_i = A_{i1}(x_1^0) \times A_{i2}(x_2^0), \qquad i = 1,2 \cdots n \tag{2}$$

Once the result of inference of fuzzy rule is generated by using Mamdani method, the general inference Q is generated using the center of area method as follows

$$Q = \frac{\sum_{i=1}^{n} W_i Q_i}{\sum_{i=1}^{n} W_i} \tag{3}$$

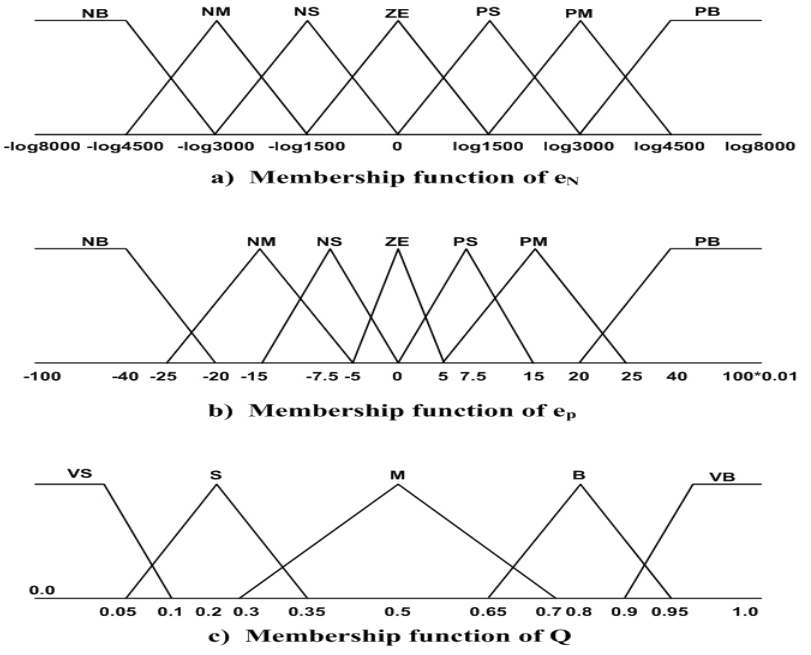


Fig. 4. Membership functions for input and output variables

We define the PI control gain using inference result Q as follows

$$K_p = Q \times k_1, \; K_I = Q \times k_2 \qquad k_1, k_2 : \text{constant} \tag{4}$$

3 Simulation

The performance of proposed fuzzy inference engine control method presented in this paper is shown by simulation using MATLAB. We use a linear model expressed with state space equation.

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t) \tag{5}$$

Where, $x_p = [x_{p1} \; x_{p2} \; x_{p3}]^T$

x_{p1} : compressor rotation speed x_{p2} : turbine inlet temperature

x_{p3} : compressor outlet pressure u_p : fuel flow

Once the maximum compressor rotating speed is 100 %, we can have the system matrix and input matrix after linearization at each point 50%, 60%, 70%, 80% and 90% as follows.

Case 50 %

$$A_p = \begin{bmatrix} -7.0893\text{E}+00 & 3.6768\text{E}+04 & 5.0478\text{E}+01 \\ 1.6640\text{E}-01 & -1.0930\text{E}+03 & 1.4680\text{E}+00 \\ 3.3399\text{E}+00 & -5.0885\text{E}+04 & -3.4759\text{E}+02 \end{bmatrix}$$

$$B_p = \begin{bmatrix} -2.3449\text{E}+04 & 3.2640\text{E}+02 & 5.9776\text{E}+05 \end{bmatrix}$$

Case 60 %

$$A_p = \begin{bmatrix} -6.9746\text{E}+00 & 2.9849+04 & 4.6627\text{E}+01 \\ 1.5664\text{E}-01 & -9.9804\text{E}+03 & 1.4095\text{E}+00 \\ 3.2662\text{E}+00 & -408540\text{E}+04 & -3.6907\text{E}+02 \end{bmatrix}$$

$$B_p = \begin{bmatrix} 3.5270\text{E}+03 & 1.2231\text{E}+02 & 5.6693\text{E}+05 \end{bmatrix}$$

Case 70 %

$$A_p = \begin{bmatrix} -6.8155\text{E}+00 & 2.6187\text{E}+04 & 4.6301\text{E}+01 \\ 1.5248\text{E}-01 & -9.2628\text{E}+03 & 1.4118\text{E}+00 \\ 3.5340\text{E}+00 & -4.9804\text{E}+04 & -3.8205\text{E}+02 \end{bmatrix}$$

$$B_p = \begin{bmatrix} -1.7240\text{E}+03 & 1.6267\text{E}+02 & 5.6502\text{E}+05 \end{bmatrix}$$

Case 80 %

$$A_p = \begin{bmatrix} -6.1171\text{E}+00 & 2.2374\text{E}+04 & 4.5137\text{E}+01 \\ 1.1650\text{E}-01 & -7.9733\text{E}+02 & 1.4274\text{E}+00 \\ 3.9098\text{E}+00 & -5.0108\text{E}+04 & -3.9057\text{E}+02 \end{bmatrix}$$

$$B_p = \begin{bmatrix} -7.5395\text{E}+03 & 1.8614\text{E}+01 & 5.5251\text{E}+05 \end{bmatrix}$$

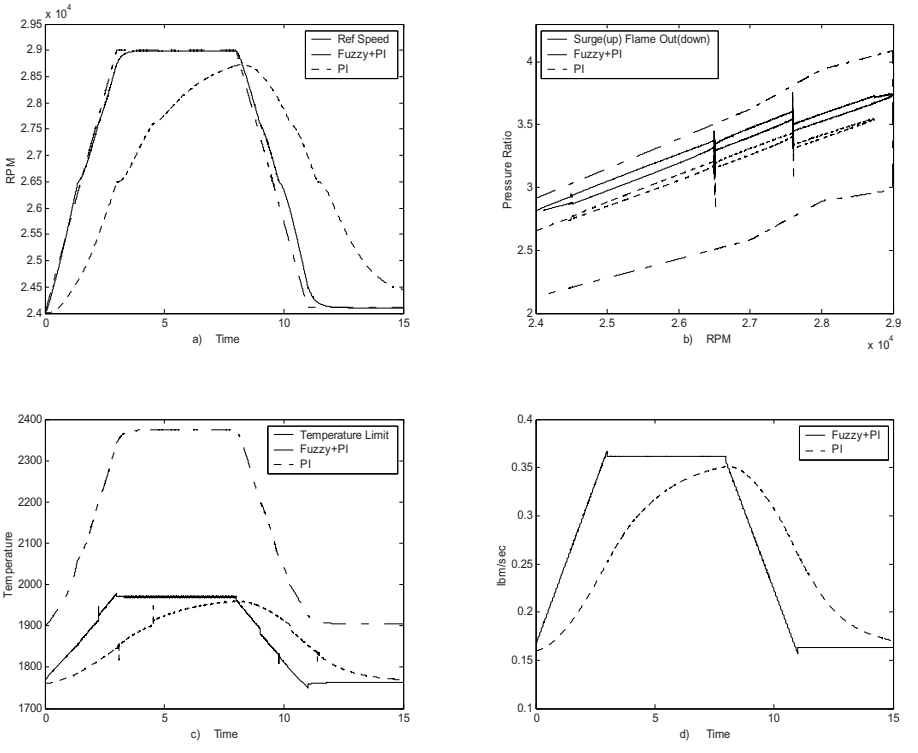
Case 90 %

$$A_p = \begin{bmatrix} -6.1182\text{E}+00 & 1.9087\text{E}+04 & 4.0209\text{E}+01 \\ 8.6668\text{E}-02 & -7.4336\text{E}+02 & 1.3260\text{E}+00 \\ 4.4379\text{E}+00 & -5.1329\text{E}+04 & -4.0410\text{E}+02 \end{bmatrix}$$

$$B_p = \begin{bmatrix} 2.0238\text{E}+04 & -4.8370\text{E}+01 & 5.5324\text{E}+05 \end{bmatrix}$$

As shown in Fig. 5 the engine adopting the fuzzy PI controller operates successfully without occurring any surge or flame out event. The rotor rotating speed at the initial state set to 24000 rpm, and accelerates by 1700rps until reaching to 29000rpm, and maintains this state for 5 seconds, and then reduces the speed by 1600rps to 24100 rpm.

The control gains of the proposed fuzzy inference PI controller are $K_p = Q \times 0.0000016$, $K_I = Q \times 0.000018$. The weigh value Q is determined depending on the control state of the fuzzy inference. As shown through computer simulation in



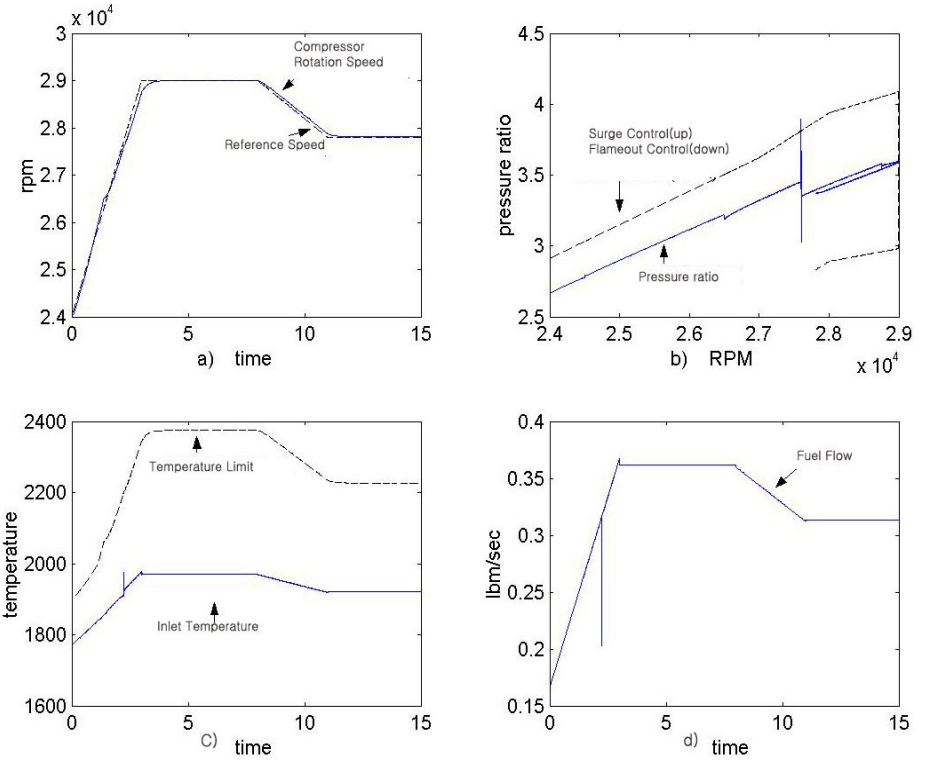
(a) compressor rotation speed (b) compressor outlet pressure ratio
(c) turbine inlet temperature (d) fuel flow rate

Fig. 5. Simulation results

Fig. 5.(a), (b) the engine run to follow the reference speed line indicated by a dashed line without exceeding both the surge control line and the flame out control line. In the Fig. 5.(a), (b), the pickings are found at 26500 rpm & 27500 rpm, however, those are one of phenomena which can occur during the rotating speed changes in the engine linear model.

Comparing its outstanding performance with PI controller shown by dotted line in Fig.5, the proposed fuzzy inference PI controller achieves much improved transient and steady state response characteristic as well as RPM & pressure ratio than the general PI controller. It also shows that the proposed Fuzzy PI controller makes the engine run to follow close to the reference speed line, and surge line.

As shown in Fig 6 (b) if the compressor rotating speed or pressure ratio exceeds the surge control line, the surge protection system using fuzzy PI controller activates by forcefully reducing the fuel flow and dropping the rotating speed under the surge control limit line. Through this simulation we show the surge protection system works as expected.



(a) compressor rotation speed (b) compressor outlet pressure ratio
(c) turbine inlet temperature (d) fuel flow rate

Fig. 6. Simulation results

4 Conclusions

In this paper we propose a turbojet engine controller of unmanned aircraft based on the Fuzzy-PI algorithm. To prevent any surge or a flame out event during the engine acceleration or deceleration, the PI-type fuzzy controller effectively controls the fuel flow input of the control system. The fuzzy inference rule made by the logarithm function of acceleration error improves the tracking error. Computer simulations applied to the linear model of a turbojet engine show that the proposed method has good tracking performance for the reference acceleration and deceleration commands.

References

1. Lehtinen, B.: Application of advanced control techniques to aircraft propulsion systems. In: First Annual NASA Aircraft Controls Workshop, Hampton, Virginia (October 1983)
2. Garg, S. : Turbofan engine control system design using the LQG/LTR methodology. In: Proc. of ACC, Pennsylvania (1989)

3. Song, Q., Wilkie, J., Grimlbe, M.J.: Robust controller for gas turbines based upon LQR/LTR design with self-tuning features. J. Dynamics Systems, Measurement, and Control 115 (1993)
4. Frederick, D.K., Garg, S., Adibhatla, S.: Turbofan engine control design using robust multivariable control technologies. IEEE Trans. on Contr. Syst. Tech. 8(6) (2000)
5. Harrison, R.A., Yates, M.S.: Gas turbine fuel control systems for unmanned applications. J. of Engineering for gas turbine and power 110, 33–40 (1988)

Implementation of QoS-Aware Dynamic Multimedia Content Adaptation System*

SooCheol Lee[1], DaeSub Yoon[1], Oh-Cheon Kwon[1], and EenJun Hwang[2,**]

[1] Electronics and Telecommunications Research Institute, Telematics/USN Research Division, Telematics Service Convergence Research Team, Daejeon, Korea
{juin,eyetracker,ockwon}@etri.re.kr
[2] Department of Electronics and Computer Engineering, Korea University, Seoul, Korea
ehwang04@korea.ac.kr

Abstract. The rapid evolution in the hardware sector brought up various end user devices like PDAs or cell phones on which online multimedia contents can be consumed. Due to different capabilities of these devices as well as individual user preferences, original multimedia resources have to be adapted in order to fit the specific constraints of such devices and to meet the user requirements. Adapting multimedia contents through user preferences is an important aspect of QoS (Quality of Service). There has been a vast amount of activities in research and standard development in this area. In this paper, we present methods for adapting multimedia contents to accommodate user preference. In addition, we propose a user preference content value model (*u-CVM*) that relates the content value of different modalities with resources. Implemented system shows that modality conversion brings a wider range of adaptation for QoS support. Moreover, the proposed approach is to be effective to apply in Digital Multimedia Broadcasting (DMB) and Telematics.

1 Introduction

In ubiquitous computing environments, users may access and interact with multimedia contents on different types of terminals and networks. Such environments include a rich variety of multimedia terminals such as PC, PDA, or cellular phone. One critical need in such a ubiquitous environment is the ability to handle the huge variation of resource constraints such as bandwidth, display, capability, CPU speed, power, and so on. This diverse set of terminal devices with different capabilities and heterogeneous networks with dynamically changing conditions, in many cases results in interoperability problems. Therefore, one major goal in today's multimedia research is the development of Universal Multimedia Access (UMA) capabilities [1], [2], [3] where a user is enabled to consume any resource, anywhere, and anytime. To achieve this

* This work was supported by the MIC, Korea, under the ITRC support program supervised by the IITA.

** Corresponding authors.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 968–977, 2007.
© Springer-Verlag Berlin Heidelberg 2007

goal, the multimedia contents have to be adapted to meet the limitations of a user terminal and network. Such multimedia adaptation could be, e.g., transcoding from one video format to another or scaling a video in the spatial domain such that it fits on the terminal's screen. Furthermore, the contents must also be adapted such that a user has an equivalent, informative experience anytime: i.e., the end point of universal multimedia consumption is the end user not the terminal.

Multimedia content adaptation is a key means for addressing the challenges of UMA [4], [5]. The general idea is to transform characteristics like network bandwidth as well as capabilities of the end user devices dynamically to meet the QoS requirements. In practice, QoS management can be done at the network level and application level as well. This paper deals with the application level, where content adaptation is an important solution to provide the QoS support.

Content adaptation [6], [7], [8], [9], [10] has two major aspects: one is modality conversion as transmoding, which converts the content from one modality to another modality: the other is content scaling, which changes the amount of resource of the content without converting its modality.

Modalities can be derived from different modes of content coding. Even different coding formats for images (GIF, JPEG) are sometimes referred to as modalities or submodalities.

There are various conditions that may affect the decision on modality conversion [7], [11]. They can be classified into four main groups: (i) the modality-presentation capability that indicates whether to display certain modalities. This can be determined from the characteristics of a terminal or the surrounding environment, (ii) user preference, which shows the user levels of interest in different modalities, (iii) the resource constraints of terminals or networks, such as the connection bit-rate, or the memory size available for the requested contents, (iv) the semantics of the content itself. For example, among a news video and a ballet video, the provider would be more willing to convert the former to a stream of text.

The MPEG-7 standard [12], [13], [14] provides an important metadata framework for describing multimedia summaries and user preference. As a result, there has been tremendous interest in building technology for UMA based on the MPEG-7 standard.

While MPEG-7 can be used to describe user content preferences for personalized filtering, searching and browsing, MPEG-21 [1] expands the scope to address seamless access of multimedia across devices, networks, and users. In particular, MPEG-21 Part 7 "Digital Item Adaptation (DIA) [3]" supports user-centric adaptation of multimedia content to the usage environment. For example, MPEG-21 DIA addresses the customization of the presentation of multimedia content based on user presentation preferences for content display/rendering, QoS, and configuration and conversion with regard to multimedia modalities. In addition, MPEG-21 DIA supports user accessibility and personalization. For example, MPEG-21 DIA facilitates adaptation of multimedia resources according to the description of user audio-visual accessibility characteristics.

In this paper, we propose a systematic approach that converts the content objects of a multimedia document according to the resource constraint and the user preference. We formulate the content adaptation process of multimedia contents as a general constrained optimization problem and then extend it to support modality conversion.

The paper organized as follows. In section 2, we describe the related work. In section 3, we present content adaptation framework. In section 4, modeling of modality conversion is shown by the *u-CVM*, which relates the content qualities of different modalities. In section 5, we present the implementation details of the system, and section 6, concludes this paper.

2 Related Work

There have been many research activities and advances in this field. Earlier work such as [11], [15], [16], [17] has explored some interesting aspects of adaptation like bandwidth reduction, format conversion, and modality replacement for Web browsing applications. Recently, international standards such as MPEG-7, MPEG-21, W3C, and TV Anytime have developed related tools and protocols to support development and deployment of video adaptation applications. The modality capability of a terminal can be determined from Usage Environment descriptions of MPEG-21 Digital Item Adaptation (DIA). MPEG-21 DIA also provides the Conversion Preference tool to specify the user preference on modality conversion, and the Universal Constraints description tool to define the constraints of the adaptation. Meanwhile, the W3C creates the Composite Capability/Preference Profile (CC/PP) protocol [18] to exchange the characteristics of users and terminals.

In order to find the conversion boundaries between modalities, [7] represented the relationship between the resource and content values of different modalities using the overlapped content value (*OCV*) model as shown in fig. 1. In the figure, *OCV* model of a video content consist of video, image, audio, and text curves. The modality curves are normally non-decreasing and saturate when the amount of a resource is large enough. Each point on a modality curves corresponds to a content version of that modality. The intersection points of the modality curves represent the conversion boundaries between the modalities. Though not really mandatory, each modality curve should cut another one, at most, at one point. Similar to rate-quality curves in image/video coding, the *OCV* model can be either operational or parametric.

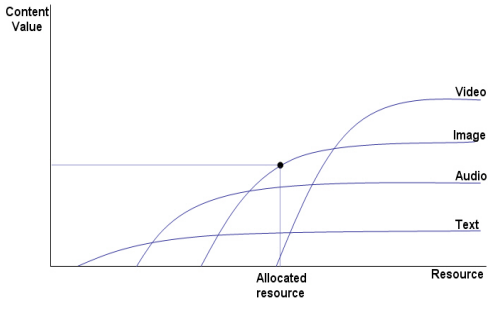


Fig. 1. OCV model

3 Multimedia Content Adaptation Framework

In this section, we present multimedia content adaptation framework. Whereas modality conversion is the main focus of this paper, we also describe content scaling in detail due to their close relationships.

3.1 Adaptation Engine

Multimedia content adaptation is carried out by an adaptation engine whose architecture is illustrated in Fig. 2. The adaptation engine consists of three main components: transmoding module, transcoding module, policy maker.

The policy maker has three important metadata inputs. The first input is the content description, which provides a format-agonistic adaptation engine with information about multimedia contents to be adapted. The second input consists of the resource constraints that show the characteristics of network and terminals. The third input is the user preference. In this paper, the user preference implicitly means the user preference on modality conversion. The policy maker analyzes these inputs and then makes optimal decisions on modality conversion and content scaling.

The transmoding and transcoding modules include specific conversion and scaling operations in order to adapt the content objects according to the instructions from the policy maker. Some objects may be passed directly to the transcoding module, without converting their modality; while others are first passed to the transmoding module, and then possibly to the transcoding module. The mechanism of a policy maker that makes accurate decision on modality conversion as well as content scaling, is the focus of this paper.

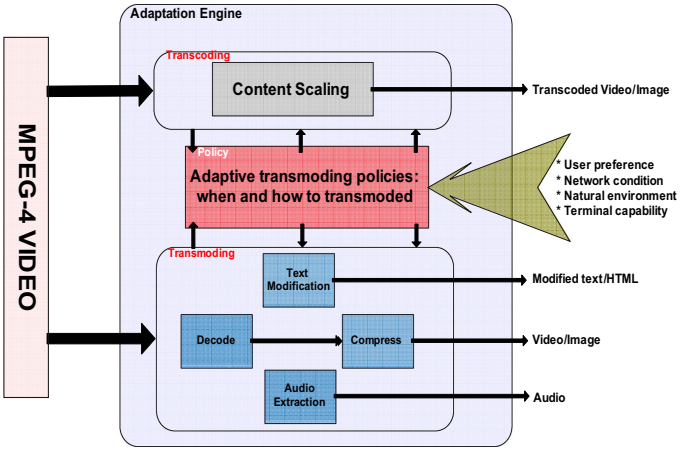


Fig. 2. Adaptation Engine

3.2 Adaptation Profile

The adaptation profile describes the terminal capabilities as well as network characteristics, natural environment, and user preference. More specific information is outlined below and the sample terminal and network description is shown in Fig. 3.

```
<DIA>
  <Description xsi type="UsageEnvironmentType">
    <UsageeEnvironmentProperty xsi type="TerminalsType">
      <Terminal>
        <TerminialCapability xs type="CodecCapabilitiesType">
          <Decoding xs type="VideoapabilitiseType">
            <Format href="urn mpeg mpeg7 cs VisualFileFormatCS 2001 7">
              <mpeg7 Name xml lanç="er">av</mpeg7 Name>
            </Forma>
          </Decoding>
        </TeminalCapability>
        <TerminalCapability xs type="DisplaysType">
          <Display>
            <DisplayCapability xs type="DisplayCapabilityType" colorCapable="true">
              <Mode>
                <Resolution horizonta ="176" vertical="144"/>
              </Mode>
            </DisplayCapability>
          </Display>
        </TerminalCapability>
      </Terminal>
    </UsageEnvironmentProperty>
    ……
    <UsageEnvironmentProperty xsi type="NetworksType">
      <Network>
        <NetworkCharacteristic xsi type="NetworkCapabilityType" maxCapacity="384000" minGrarnateed="32000"/>
        <NetworkCharacteristic xsi type="NetworkCapabilityType">
          <AvailableBandwidth maximum ="2560" average="8000"/>
        </NetworkCharacteristic>
      </Network>
    </UsageEnvironmentProperty>
  </Description>
</DIA>
```

Fig. 3. MPEG-21 DIA

a) *Terminal capabilities*: User can adapt various formats of multimedia for viewing on a particular terminal by the terminal capabilities such as codec and I/O capabilities. Codec capabilities specify the format that particular terminal is capable of encoding or decoding. I/O capabilities include a description of display characteristics, audio output capabilities.

b) *Network characteristics*: Network capabilities and network conditions affect multimedia adaptation for transmission efficiency. For instance, we can lower the network bandwidth of a video stream if the available bandwidth is insufficient. Network capabilities define the maximum capacity of a network and the minimum guaranteed bandwidth and network conditions describe network parameters such as the available bandwidth and delay characteristics.

c) *Natural environment*: Natural environment pertains to the physical environmental conditions around the user such as lighting condition or noise level, or a circumstance such as the time and location.

d) *User characteristics*: Usage characteristics include a description of user information, usage preferences and history. User preferences define preferences related to the user hardware such as PDA, handheld PC. For audio, the specification describes preferred audio power and equalizer settings. For visual information, the specification defines display preferences such as the preferred color temperature, brightness, saturation, and contrast.

4 User Preference Content Value Model

During the adaptation, there can be many possibilities of modality conversion, but the user may prefer some modalities to others. The role of modality conversion is to

enable the user to specify his/her choice on the modality. User preference should not allow for fixed choices of modality conversion only. For example, for a user request that all videos be converted to audios, yet if the terminal cannot support audio modality, the request will be discarded.

The user preference on a conversion is divided into two cases: qualitative order and a quantitative weight. Given an object, the order of a conversion from the object's original modality indicates the order by which the corresponding destination modality will be selected as the amount of resource is reduced. As described in section 2, with a given *OCV* model, there are already existing orders of conversion, which can be used for the default orders of the user. Using the user preference modality conversion, the user may specify new orders of conversions that are different from the existing orders. In the paper, the *OCV* model will be extended by the user preference order.

The weight of a conversion indicates the importance, in the user's, of the corresponding destination modality of an objects. In this paper, we focus on the user preference in quantifying the content value.

4.1 *u*-CVM by the Order of Conversion

When a modality is not supported, the versions of the modality have zero content value at the terminal. Thus the curves of unsupported modalities should be removed. Various characteristics of terminals, user, and environments, which may affect the set of supported modalities, have been standardized as the description tools in MPEG-21 DIA. Denote U_i as the set of supported modalities of object U_i, $= M \cap \{1, \ldots, J_i\}$. M is the modality capability that lists the indexes of supported modalities. The content value V_i is now represented by:

$$V_i = \text{Max}\{VM_{ij}, j \quad U_i\}$$

The orders of conversions represent the qualitative preference of the user. The user may just want to specify this qualitative preference level, ignoring the weights of the conversions which are quantitative. As mentioned in section 2, the *OCV* model of a given object has the existing orders of conversions. If the user's order of conversions is different from the existing orders, the *OCV* model will be modified.

Let O_e (j) and O_u (j) be the existing order and user preference order of the conversion from the object's original modality to modality j, respectively. The algorithm to the user preference order of conversions can be carried out as follows:

Step 0: $j = 1$
Setp 1: $j^* = 1$
Step 2: if $j^* \quad j$ and $\{j^*, j\} \subset U_i$, go to Step 3, otherwise go to Setp 5.
Setp 3: if $O_e(j) > O_e(j^*)$, go to step 4, otherwise go to step 5.
Step 4: if $O_u(j) < O_u(j^*)$, remove VM_{ij} and go to Step 6, otherwise go to Step 5.
Step 5: $j^* = j + 1$, if $j^* > J_i$, go to Step 6, otherwise go back to Step2.
Step 6: $j = j + 1$, if $j > J_i$, stop, otherwise to back to Step1.

Fig. 4. Modality conversion algorithm by user preference

Basically, this algorithm is considered to remove modality curves whose orders are not compatible with the user preference. Normally, if O_e (j) is higher than O_e (j*), modality j will be selected before modality j*. However, when considering the user preference, if O_u (j) is lower than O_u (j*), VM_{ij} should be removed so that modality j will not be selected before modality j*. This comparison process is repeated for every modality in the *u-CVM*.

As an example, given the *OCV* model in Fig. 1, we have the existing orders of conversions as follows: O_e(1) of video to video is the first, O_e (2) of video to image is the second, O_e (3) of video to audio is the third, and O_e (4) of video to text is the fourth. By this order, the audio modality will be selected before the text modality. Assume that the user preference order of conversions are specified as follows: O_u (1) is the first, O_u (2) is the second, O_u(3) the fourth, and O_e (4) is the third. That means the user prefers the text modality to the audio modality.

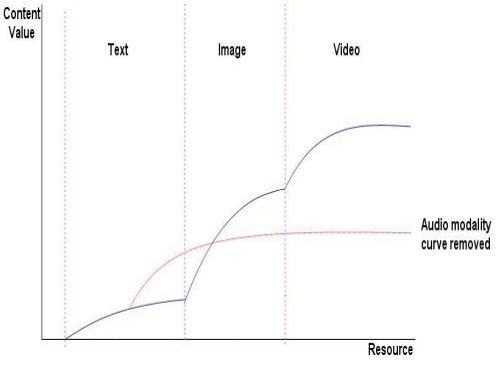


Fig. 5. *u-CVM* graph

The above algorithm first takes the video modality to check against other modalities as in Step 3 and 4; and the decision is not to remove the video modality curve. Next, the decision is similar for the image modality. However, in order not to select the audio modality before the text modality, the audio modality curve is removed by Step 4. Finally, the *u-CVM* is shown in Fig. 5.

5 Implementation

We have implemented a prototype system to demonstrate the feasibility and efficiency of the adaptation engine. The adaptation engine selects an optimal adaptation modality based on content value and user preference, and carries out adaptation of an MPEG-4 video. Fig. 6 shows the user interface.

According to the user preference, the requested video is transmoded and displayed into different formats such as different frame rate and frame size. By monitoring the network condition, we can obtain the terminal and network profile for each modality.

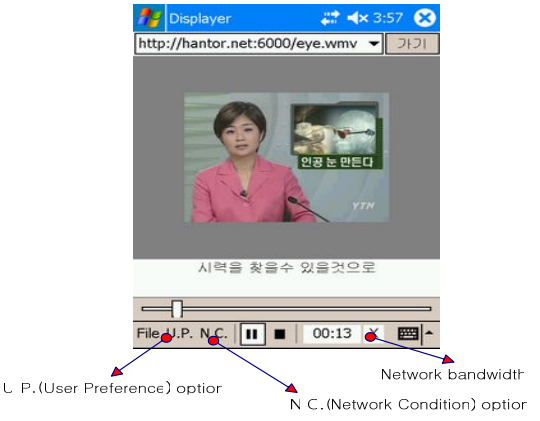


Fig. 6. User interface

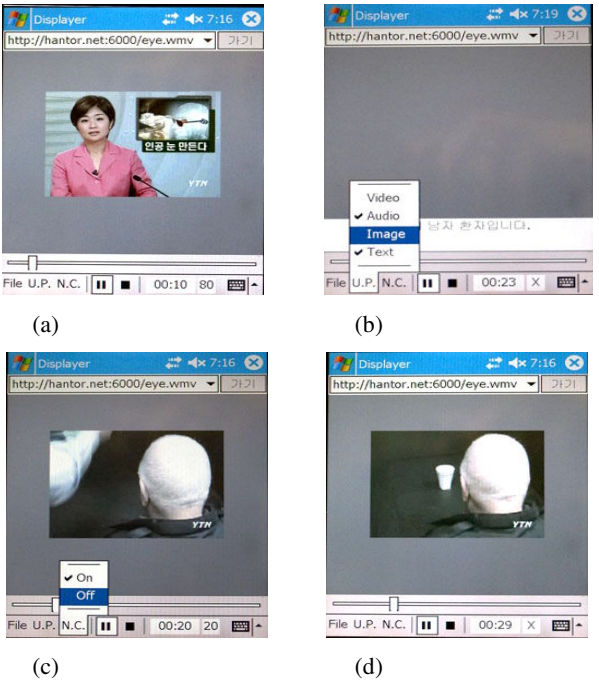


Fig. 7. Adaptation process scenario

Then the collected adaptive parameters can be fed to the adaptation engine for determining any degradation. When network bandwidth consumption is saturated, the PDA client can display the video. However, the adaptation engine starts degradation by transmoding under the scarce bandwidth. During this degraded situation, the user can still listen to the audio with images through key-frames or read the text (closed

caption). Fig. 7 illustrates the scenario of adaptation process using PDA. A user first input URL and clicks on the "go" button to get an AV clip. This clip is then played on the center of the video playback window. User can then select modality and network constraint by choosing U.P and N.C. options. When the N.C option enabled, network bandwidth is indicated as shown in (c). (b) is illustrate the various combination of U.P options as audio + text.

6 Conclusion

In the coming ubiquitous computing environment, the same multimedia content may have to be transcoded and transmoded in different forms according to network capabilities and user preference. Content adaptation is a major issue that needs to deal with for developing an adaptive video delivery system. In this paper, we proposed *u-CVM* which is an extended overlapped content value (*OCV*) model to represent the dependence of content value and user preference. We also design the adaptation engine and implement the framework for adaptation driven user perceived quality rather than just bit-rate reduction. The system developed in this paper can also be applied to other scenarios of adaptation, for instance, adaptation of scalable video streams and optimal adaptation based on a subjective quality measurement.

References

[1] MPEG Requirements Group.: MPEG-21 Multimedia framework, Part 1: Vision, technologies and strategy, Proposed Draft Technical Report, 2nd edn., Doc. ISO/MPEG N6269, MPEG Waikaloa Meeting, USA (December 2003)
[2] MPEG MDS Group.: Multimedia framework Part 7: Digital item adaptation, Final Draft International Standard, Doc. ISO/MPEG, N6168, MPEG Waikaloa Meeting, USA (December 2003)
[3] Vetro, A., Timmer, C.: Digital Item Adaptation: Overview of Standardization and Research Activities. IEEE Trans. Multimedia 7(3) (2005)
[4] Mohan, R., Smith, J., Li, C.-S.: Adapting multimedia internet content for universal access. IEEE Trans. Multimedia 1(1) (1999)
[5] Lei, Z., Georganas, N.D.: Context-based media adaptation in pervasive computing. In: Proc. of IEEE Canadian Conf. on Electrical and Computer (CCECE'01), Toronto (May 2001)
[6] Nam, J.H., Ro, Y.M., Huh, Y.S., Kim, M.C.: Visual Content Adaptation According to User Perception Characteristics. IEEE Trans. Multimedia 7(3) (2005)
[7] Thang, T.C., Jung, Y.J., Ro, Y.M.: Modality conversion for QoS management in universal multimedia access. IEE Proc.-Vis. Image Signal Process 152(3) (2005)
[8] Fox, A., Gribble, S.D., Brewer, E.A., Amir, E.: Adapting to network and client variability via on-demand dynamic distillation. In: Proc. 7th Int. Conf. Arch. Support Prog. Lang. Op. Sys (ASPLOS-VII), Cambridge, MA (1996)
[9] Fox, A., Gribble, S.D., Chawathe, Y., Brewer, E.A.: Adapting to network and client variation using active proxies: Lesson and perspectives. IEEE Personal Commun. 40 (1998)
[10] Vetro, A., Christopoulos, C., Sun, H.: Video Transcoding Architectures and Techniques: An Overview. IEEE

[11] Ma, W.Y., Bedner, I., Chang, G., Kuchinsky, A., Zhang, H.J.: A framework for adaptive content delivery in heterogeneous network environments. In: SPIE Multimedia Computing and Networking, San Jose, California, USA (2000)
[12] ISO/IEC JTC1 SC29/WG11 N3465, MPEG-7 Multimedia Description Schemes WD, Beijing (2000)
[13] Ramanujan, R.S., Newhouse, J.A., Kaddoura, M.N., Ahamad, A., Chartier, E.R., Thurber, K.J.: Adaptive Streaming of MPEG Video over IP Networks. In: Proc. of the 22nd IEEE Con. on Computer Networks (LCN 97) (1997)
[14] Magalhaes, J., Pereira, F.: Using MPEG standards for multimedia customization. Signal Processing: Image Communication 19, 437–456 (2004)
[15] Snoek, C.G.M., Worring, M.: Multimodal video indexing: A review of the state-of-the-art. Multimedia Tools and Applications (2004)
[16] Halal, S., Sampath, L., Birkett, K., Hammer, J.: Adaptive delivery of video data over wireless and mobile environments. Wireless Communications and Mobile Computing 3, 23–36 (2003)
[17] Han, W.Y., Kwon, O.C., Park, J.H., Kang, J.-H.: A Gateway and Framework for Telematics Systems Independent on Mobile Network. ETRI Journal 27(1) (2005)
[18] W3C: Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0 W3C Recommendation (2004) http://www.w3.org/TR/2003/REC-CCPP-structure-vocab-20040115/

Experience of Efficient Data Transformation Solution for PCB Product Automation

Jung-Soo Han[1] and Gui-Jung Kim[2]

[1] Division of Information & Communication, Cheonan University,
115, Anseo-Dong, Cheonan, Chungnam, Republic of Korea
jshan@cheonan.ac.kr
[2] Department of Bio Medical Engineering, Konyang University,
26, Nae-Dong, Nonsan, Chungnam, Republic of Korea
gjkim@konyang.ac.kr

Abstract. This paper embodies Data Transformation System for PCB (Printed Circuit Board) production automation. Data designed by CAD cannot be compatible with data of PCB production line, so it should have a data compatible automated system. This paper has been developed a human error extraction algorithm for PCB production automation and an algorithm that transforms reliable data for PCB production line and also an interface for this. As the result, processing rate has been improved and it helps PCB production automation with reliable optimized data.

1 Introduction

This paper shows how to automate data transformation in PCB production. CAD data designed is not compatible with data of PCB production line in PCB production. That is, it is produced through equipments such as Loader, Surface Mount, Clean Machine, Screen Printer and so on[1][2]. However, data designed by CAD is not compatible with these equipments, so it is the actual condition that data for each PCB is operated by manual work. In this process, human errors are happened, poor products are frequently happened and it costs a lot. It is hard because it causes low reliability [9]. One manual process takes about 8 hours to only convert data for a PCE production. It takes about 1 to 2 days including test process. The more kinds of PCB, the more money and higher cost are needed exponentially. Therefore, this paper aims the compatible data automation through data analysis of each item designed by CAD from the PCB assembly process.

2 Automated Data Transformation System

2.1 System Architecture

Data designed by CAD have data that decide to be assembled to the front side or the back side of the board and also have data of total numbers of items of the

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 978–986, 2007.
© Springer-Verlag Berlin Heidelberg 2007

board or the features of the items [3]. However, these data are not compatible with PCB production line. That's why CAD designer designs it according to his/her own design method without well-understanding it. Therefore, the PCB manufacturer has to transform this information into reliable data by manual work. This inefficient work has a direct connection with high cost and low productivity of PCB manufacturer. And also human errors that are possible to be happened from manual work could cause a huge amount of loss. Fig.1 shows how to extract item information through information analysis of CAD data. By analyzing whole process of PCB production line, PCB item information needed is achieved and data will be classified each item [3]. Make sure both items information which are analyzed from CAD and from PCB line are exactly the same. If they do not, they should be stored in repository after correcting to have exact item information. The information that are constructed in repository includes data such as item names, item numbers, locations, angles, standards, amounts and so on. Now the repository identifies keywords for item search through interface and completes to be ready for data transformation program. In this process, human errors happen during CAD data error correction. To solve this problem, data should be verified by using human error extraction algorithm. If there is another error, it will be informed inter-face user [4][5]. And data with error among CAD data will be input again. This item information comes from data which is given from analyzing PCB production line process. While CAD designer decides PCE board coordinates on his own, PCB production line computes coordinates with the starting point as an angle of the board. So, coordinate values of CAD data will be transformed into reliable coordinate values for PCB production line. The transformation of coordinate values shows user how to design present coordinate values of CAD data by deciding where the starting point should be. Fig.2 shows the stage how to remove CAD data error by CAD designer or Human error by storing extracted CAD data in data repository [6]. It is to store data with no error in the repository in this stage.

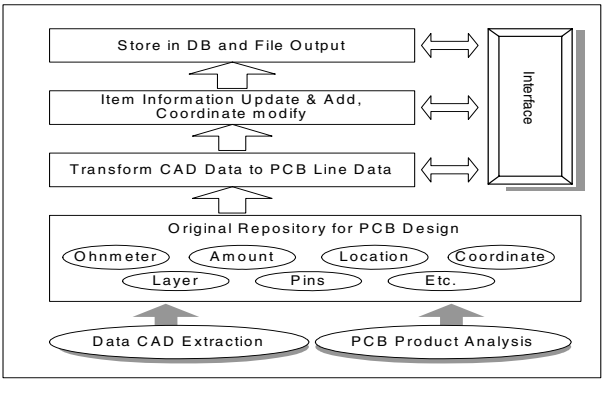


Fig. 1. Architecture of Database Transformation System

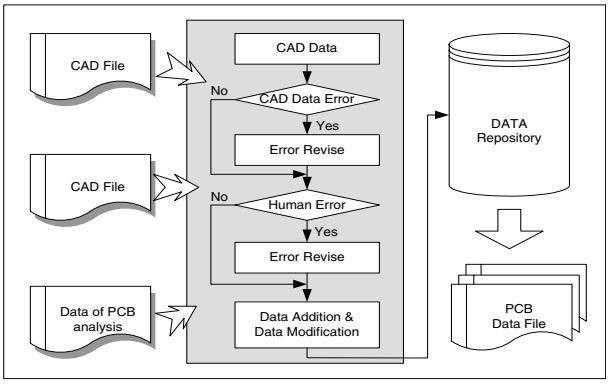


Fig. 2. Conversion Reliable Data for Repository

2.2 Data Error Extraction and Sort

Error extraction and correction of CAD data themselves should be done in order to convert CAD data into PCB data. There is a possibility that CAD data includes Human error at any time because CAD data is designed by CAD designer. For this kind of error extraction, verification algorithm has been developed. Verification algorithm considers CAD data as a Human error when their item names, item numbers, locations, standards, amounts etc are not confirmed to stored information earlier by PCB production equipments. Also it will have new values as inputs in those cases as followed such as when there is no confirmation of item name and item number among CAD data, when quantities that can be accepted by PCB are overloaded, when there is a difference about numbers of location information and amounts of items, and so on. The information that

```
Input CAD data in Database
Database field are item_name, item_number,
Location_NO, coordinates, amount
save each filed of Location_NO data
For( All Q'ty )
if( # of Q'ty same Location_Number)
make records for # of Q'ty in database
separate Location_NO in each number
Input standard radian
FOR(all items)
IF(item_number != amount)
output ("Human Error")
input new item_number
ENDIF
```

Fig. 3. Data Error Verification Algorithm

Specification	Q_ty	Location_No
DSV-830(MT1389) for LG	10	C104-107,C110,Q6,9-12

CAD Information of Database

Specification	Q_ty	Location_No
DSV-830(MT1389) for LG	1	C104
DSV-830(MT1389) for LG	1	C105
DSV-830(MT1389) for LG	1	C106
DSV-830(MT1389) for LG	1	C107
DSV-830(MT1389) for LG	1	C110
DSV-830(MT1389) for LG	1	Q6
DSV-830(MT1389) for LG	1	Q9
DSV-830(MT1389) for LG	1	Q10
DSV-830(MT1389) for LG	1	Q11
DSV-830(MT1389) for LG	1	Q12

Transformation Result

Fig. 4. Sorted Data

is input is exchanged established information at the item repository. However, there is another possibility that there input values are wrong, so they are needed to be verified exact data through data error verification algorithm. If they do not include any error, then they will be transformed through sort algorithm which is next step. Fig.3 shows data error verification algorithm. Sort algorithm classifies each item after storing data in DB. In the case of Location_Number, there is a little location information within a field of database. So, it classifies each of location information and stores in the new fields, then it will confirm that there is an agreement between Q'ty and Location_Number[7][8]. If there is no agreement, new Q'ty and Location_Number should be input. Fig.4 shows new database transformed by sorting Location-Number data stored in the database. Finally, it compares the number of items and the total amount.

2.3 Item Information Correction and Addition

It corrects and adds item information to compare to CAD data and the data of PCB items through analyzing whole PCB production line. To do so, it first looks for the same data by comparing Location_No field between two Tables in the database. The values of Location_No have the information about which location should be IC item assembled on PCB board. If there is the same data after comparing two Tables, it stores Specification and Q'ty of Table1, and Orient, X, Y, and Layer of Table2 onto another Table in the database. The term "Orient" means an item angle of CAD data. Surely the values of Location_No are stored together. Because the data which should be input into final PCB production line includes Location_No, X, Y, Orient, Specification, and Layer. That is, these values are the minimum information that shows what item should be fit which location on actual board. In the case of angle of CAD data, it is possible to be given any angle value by CAD designer. Because CAD designer makes out CAD by using item standard angle without analyzing item angle of PCB production line. Therefore, this paper analyzes the angles that standard angle for these

angles and the angle wanted by PCB production line by comparing data analyzed the whole process of PCB production line earlier. If these values are not the same, it will correct and transform like this, data stored into a Table are ready for the stage for coordinate correction and Layer separation.

2.4 Coordinates Correction and Layer Separation

PCB item-by-item data of the data stored into a Table include data like CAD, but it is to be sure that each coordinates is designed with which location as the center because coordinate value (x,y) of CAD data and coordinate value (x,y) of the PCB production line are made different system from each. CAD designer decides the stating point as he puts coordinate value into about the middle of location of PCB board. On the other hand, a corner of PCB board

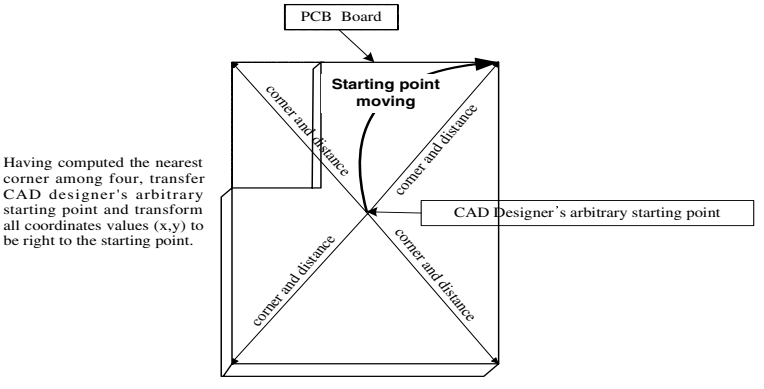


Fig. 5. Starting Point Correction

```
Location_No save in Database Table1
Location_No, X, Y,  Orient save in Database Table2
FOR(Last Recode)
  IF (Location_No of Table 1 =Location_No of Table 2)
      Table 1 JOIN Table2
  END IF
+-+-  Input standard (X,Y)
  conversion (X,Y)
  IF (CAD_data(radian) != Real Radian)
      input new standard Radian
      Do radian transformation
  ENDIF
  IF (Layer = top)
      save of Top_Table
  ELSE
      save of Bottom_Table
  ENDIF
```

Fig. 6. Data Transformation Algorithm

in the PCB production line is used as the starting point. So, find out which location that CAD designer decided as the starting point and then compute the distance between the nearest corner among four of PCB board and the starting point through the starting point algorithm. Having done that, the start-ing point algorithm will ask user to put the distance error via interface and will correct all coordinate values. Fig.5 shows that it computes any of the starting points of CAD designer and move. Coordinate values will be stored from the corrected data to another new Table according to Layer values. That is why Layer values have the value which can decide to assemble IC item onto the front or the back of PCB board. If the value of the Layer is Top, it will be assemble onto the front. If it is Bottom, it will assemble onto the back. So, they will be stored separately. For omitted data, user will be informed to input Layer value and new Layer value will be input. It is because PCB production line is divided Top and Bottom. Also production line inputs data as a text file, it is also to be input as a file according to Layer value. Fig.6 shows the Data Transformation Algorithm. This algorithm first puts Table1 and Table2 into one Table, and then computes. The table will be achieved data transformation through coordinate correction and Layer separation.

3 Test and Performance Estimation

3.1 Database and Output

Fig.7 shows output information. It reads necessary item data for the final line of PCB production and outputs the final text file. When it output the text file, it outputs only one item by one line. Output data of item information ore ordered Location_No, Specification, Orient, X, Y, and Layer. Each value is divided by a tap interval. In the way of design interface, users should approach easily to data stored in the database in for data verification or conformation. it is not only to prevent human error by CAD designer's wrong design, and also to correct easily data which need a correction. Fig.8 is the result of data transformation system that is presented by this paper and Table 1 shows the process performed by each step.

Table 1. Process of Each Step

Each Stage	Explanation of Processing Performed by Each Step
Step 1	Table sort, error correction
Step 2	Item information correction and addition
Step 3	CAD designer's the starting point conformation
Step 4	Correction of the starting point, X, Y coordinate values
Step 5	Database recording
Step 6	Details conformation
Step 7	File output

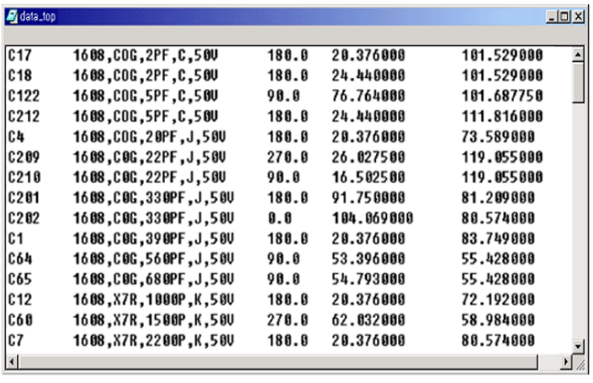


Fig. 7. Final Output Result (Top Layer)

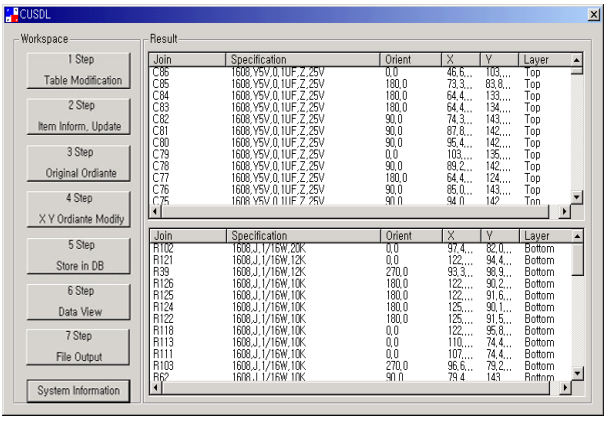


Fig. 8. Data Transformation Results

3.2 Comparison to Established System

We have compared the system presented by this paper to established data transformation system SMT[2]. Comparison standard was the stage that transformed cad data which has more than 500 items into reliable data for PCB production line. We have transformed thirty-five times. Comparison standard estimates the performance of software system according to how easy and comfortable to use and how much help to perform for users. The estimation depends on the increase of efficiency, improvement of productivity, decrease of error, decrease of training and increase of user accommodation. Increase of efficiency estimates the method of users' performance if it is the same as system interface. Reduce of errors is because of wrong user interface, it estimates if it avoids contradiction

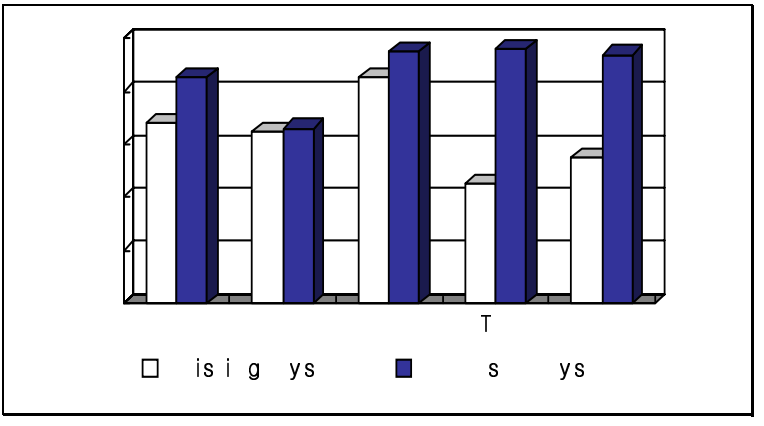


Fig. 9. Comparison to Established System: EI: Efficiency Increment, PE: Product Enhancement, ED: Error Decrease, TD: Training Decrease, UI: Usability Increment

and ambiguity like it reduces errors. Decrease of train-ing estimates how easy to learn interface. Presented items takes 100 as satisfaction, total eight of users were required. Fig.9 is the graph that shows the comparison be-tween established system and the system presented by this paper.

4 Conclusion

This paper aims that users easily and conveniently transforms CAD data into reliable data for PCB production line. In the meantime, this process has taken about 1 to 2 days by manual work and it has to be passing the test by eye directly. That is why there is a possibility to take a lot of time and to produce poor. This paper has gotten rid of minimum error by doing that verifies human error in the CAD data and com-pares data analyzed the whole PCB production. It transforms data with removed error into reliable data for PCB production line through transformation algorithm. There-fore, factory automation has been completed through data compatible among distinct machines for the exact and efficient PCB production.

References

1. http://www.qmaxtest.com
2. http://www.smtkorea.co.kr
3. Han, K.H.: PCB Design Package Program CAD Star. SoeWun Publishing, Korea (1999)
4. Debbie, S., Caroline, J., Mark, W., Shailey, M.: User Interface Design and Evaluation. Morgan Kaufmann Pub, San Francisco (2005)
5. Lauesen, Soren.: User Interface Design. Addison-Wesley, Reading (2004)

6. Dietrich, Suzanne, Susan, D., Urban.: An Advanced Course In Database Systems. Prentice Hall, Englewood Cliffs (2004)
7. Ingraham, Laura, R.: Data Modeling And Database Design In An Access Environment. John Wiley & Sons Inc, Chichester (2005)
8. Zongmin, M.: Advances In Fuzzy Object-Oriented Databases. Idea Group Pub, USA (2004)
9. Peters, J.F., Pedrycz, W.: Software Engineering. John Wiley & Sons, Chichester (2000)

Performance Evaluation for Component Retrieval

Jung-Soo Han

Division of Information & Communication, Cheonan University,
Anseo-Dong, Cheonan, Chungnam, Republic of Korea
jshan@cheonan.ac.kr

Abstract. In this paper, we proposed enhanced spreading activation algorithm. Because SARM requires a lot of computation time, enhanced SARM is possible as reducing unnecessary activation value. By removing the parts with a small number of references in the retrieval process, the time taken for retrieval in simulation has been reduced by 37.8% in comparison with SARM. Also, in this paper, we focus performance evaluation of our component retrieval system.

1 Introduction

Software reuse is a technique of reducing developing time and cost and enhancing the quality and productivity by reusing the software library commonly witnessed in many cases of system development. Sharing the information and development method of the system, it also reduces the burden of development on the part of the developer, and the concentration on the expandability of the software and its maintenance en-hances the reliability, usability, and efficiency of the software [1]. With the development of object-oriented methodologies such as OOP(Object-Oriented Programming), OMT(Object Modeling Technique), and OOA/OOD, the technology for reuse has been concentrated on the efficient management and methods of reuse. For reuse, how-ever, techniques for efficiently storing, managing, and retrieving objects are required. Software components, through the analysis of existing software, are reused after deriv-ing, classifying, and storing the reusable components and their information, i.e., the code, class, inheritance relationship, and diagram information in the repository [2,3,6,8]. The existing reuse system, depending on the component retrieval for object reuse and document information, has been designed on the basis of limited experiences on the actual reuse. This paper constructs the repository for the object-oriented reuse with the information of the software components-class name, member function, attribute, and inheritance. The Retrieval method of Spreading Activation [4,5], used for this study, is based on the connection relaxation of the connectivity between components. This technique also enables the retrieval of components not directly connected, though related to the retrieval term. The problem is that it takes much time because it measures the similarity using the activation value. This paper is an improvement in that it applies the computing-by-selection

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 987–996, 2007.
© Springer-Verlag Berlin Heidelberg 2007

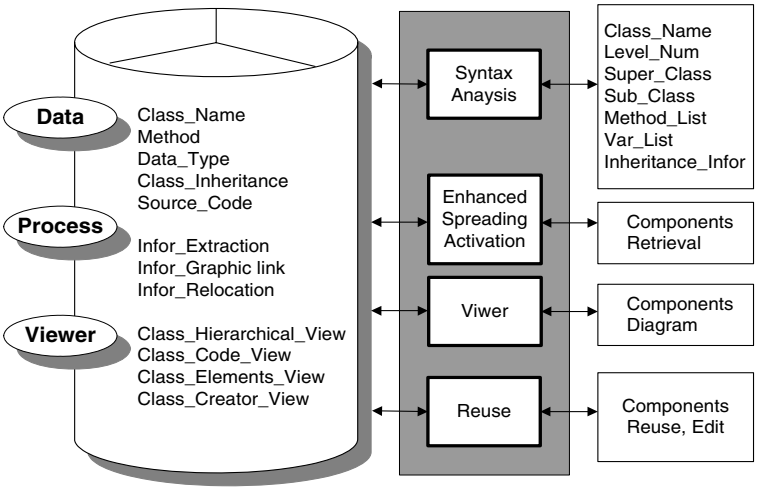


Fig. 1. Software Components Reuse System

method maintaining the result of expanded retrieval and reduces the number of computation of activation value. With the input of the query, the components are obtained through E-SARM (Enhanced Spreading Activation Retrieval Method) and their similar sets are retrieved in order of their activation value. Thus this paper makes it possible to retrieve similar components as well as components through query with E-SARM algorithm. This paper is organized as follows. In Section 2, we describe the structure of the component retrieval system and the enhanced SARM algorithm. Section 3 evaluates the performance of the enhanced SARM and compares it with other systems. Finally, Section 4 concludes the paper and provides an outlook future work.

2 Component Retrieval System

2.1 Component Retrieval System

The system for the software component reuse saves the information for components in the repository through the syntax analyzer, as shown in Fig.1. The repository consists of Data, Process, and Viewer; when a query is input though the retrieval viewer; the components appear as a list in order of the activation value after the activation values are computed through E-SARM. For the reuse of a component, the information of the component, selecting the component, and using the copy and paste function automatically transmit its source code information, and the graphic information. But as the movement of components within the hierarchy diagram means the change into a subclass within a new class, the information concerning the inheritance relationship of the source code of the moved class should be transformed. In other words, the name of the superclass, which means the inheritance, should be altered[7]. This can be achieved

by altering the information of inheritance of the subclass by reference to the name of the superclass in inheritance relationship. Modification for the reusable components is necessary for each of the functions of new software even if the components are moved on the hierarchy diagram. The present system makes the modification of the components possible by providing an editor. The editor also sup-plies the functions of methods and attributes for the modification of components. The selection of the class to modify on the class hierarchy diagram activates the editor with the source code and after modifying and saving the code, the code information of the component is altered in the repository and the reuse of the component is achieved.

2.2 Component Analysis

Components for reuse form a repository containing the information on the name of the class, the member function, properties, variables, the inheritance relationship be-tween the superclass and subclass extracted through syntax analysis. The information of class extraction, the information of link based on the inheritance information, and the location information of each class are collected for the representation of the hierarchy diagram for each of the components. The location information is extracted ac-cording to the label within the hierarchy diagram as in the hierarchy diagram the location information of each class should be represented equivalently with the superclass at the center.

2.3 Component Viewer

Viewer is a function for showing the analyzed information of the components as a graphic, represented as an object-oriented diagram using the number of subclasses in inheritance relationship with each class, with the topmost class at the center. The schema information is represented as a diagram after designating the object and inheritance relationship by computing the depth and width of the object, the numbers of classes and subclasses, and the number of classes in each label. The name of each class is shown along with its location. The understanding of the components is enhanced by the code information link, which shows the source code for each class on the diagram. Accordingly, the viewer makes it possible to view the diagram, class inheritance relationship, and the code information of each class using the component information saved in the repository. The important function of the viewer rearranges the increased or decreased classes in each label by reallocating the location of each class according to the number of increase or decrease of the classes in the process of insertion or deletion of the whole components or a portion of classes.

2.4 Enhanced SARM

In SARM the repetition of the cycle for computing the activation value is done for retrieving similar components. Such a method takes much time as the activation value of each of the queries and components influences the computation of the activation value of other queries and components. To solve such a problem,

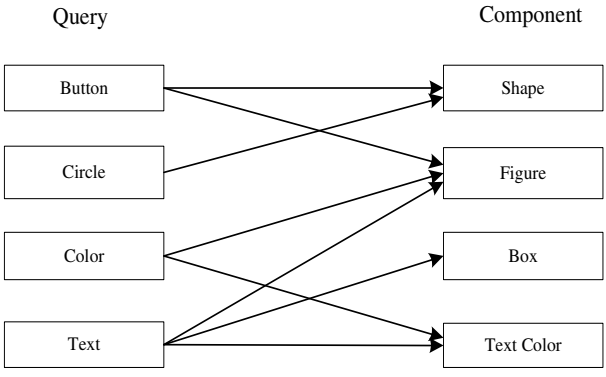


Fig. 2. C1ass and Query Relationship

Table 1. E-SARM Parameters

Symbol	Definition
$D_i(t+1)$	activation value of component D_i in time (t+1)
$\delta_{D_i}(t)$	activation value in time t by decay rate
$\phi_{D_i}(t)$	number of inputs in component i
θ_D	component decay rate = 0.1
θ_T	query decay rate = 0.25
M	maximum activation value = 1.0
m	minimum activation value = -0.2
D	D is component
T	T is a query

this paper has shortened the retrieval time by removing the link information of components with small indexing and thus, reducing the number of computation of the activation value. E-SARM (Enhanced Spreading Activation Retrieval Method) allows retrieving only components with greater connection by removing the link information of queries or components that fall behind the standard after a certain times of repetition of cycle, thus reducing the expansion range of the queries.

Fig. 2 shows the connection relationship between the queries and classes. Class is defined as the topmost class, which is equivalent to the root node of the software component. With the input of the query Button, three classes are retrieved. Here the query Button is connected to the classes, Shape and Figure, but not to classes Text_Color and Box. However, we can see that two classes (Box, Text_Color) are connected through Button(query) → Figure(class)→Text_Color (class), and then Button(query)→Figure(class)→Text(query) →Box(class). The connection to Box, how-ever, is removed because it is seldom referred to in the retrieval process. Thus, the activation value is computed referring to the connected nodes of each query and class components. The value for Shape is

higher than that for Figure because the reference number for Shape is higher than that for Figure. The comparison between the existing SARM and E-SARM in the simulation of the average reference number of components shows that the number of computation has been reduced by 37.8%. Table 1 shows the definition of variables for activation value and the activation value of the queries and components is computed using the metrics in algorithm. As the cycle is repeated, the activation value becomes stable and the computation process is finished with the removal of the parts whose reference number falls under the standard.

Enhanced Spreading Activation Algorithm

Repeat
 If $\forall$ *relationship and* $\forall$ *index*
  push *Current_index in Stack*
Until *query or component number*
increase *query_visit_count*
increase *component_visit_count*
If *first query*
 initialize *query_act_value := 1.0*
if *a query*

$$D_i(t+1) = \begin{cases} \delta_{D_i}(t) + \phi_{D_i}(t)(M - D_i(t)) & \text{if } \phi_{D_i}(t) > 0 \\ \delta_{D_i}(t) + \phi_{D_i}(t)(D_j(t) - m) & \text{if } \phi_{D_i}(t) \le 0 \end{cases}$$

$$\delta_{D_i} = (1-\theta_D) D_i(t)$$

Else if *a component*

$$T_j(t+1) = \begin{cases} \delta_{T_j}(t) + \phi_{T_j}(t)(M - T_j(t)) & \text{if } \phi_{T_j}(t) > 0 \\ \delta_{T_j}(t) + \phi_{T_j}(t)(T_j(t) - m) & \text{if } \phi_{T_j}(t) \le 0 \end{cases}$$

$$\delta_{T_j} = (1-\theta_T) D_i(t)$$

Else *ErrMsg := "Not Calculate Activation Value"*
 Procedure call pop(*Get_Pos*)
 If (*Not MAX_CYCLE*) **increase** *cycle*
 If *(cycle > MAX_CYCLE)* **break** /*MAX_CYCLE=3~5*/
 If *cycle is between 2 & 3*
  Repeat
   If *query_visit_count or component_visit_count := current_cycle_count*
    set *AvgVisit := AvgVisit + VisitNum[]*

2.5 Component Retrieval

Components directly connected to the query and similar ones indirectly connected to it can be retrieved through Enhanced Spreading Activation algorithm. Every data is saved in the repository and such a process is repeated until the activation value connected to the query is stabilized with the completion of computation [9]. The topmost class of the whole components, the Base Class is retrieved within the standard range of the activation value. With the increase of reference the activation value increases and the similar components are also retrieved through the reference function. With such retrieval the options for selecting the reusable components expand as there are many components provided, and the understanding of the components in terms of the query is enhanced. In

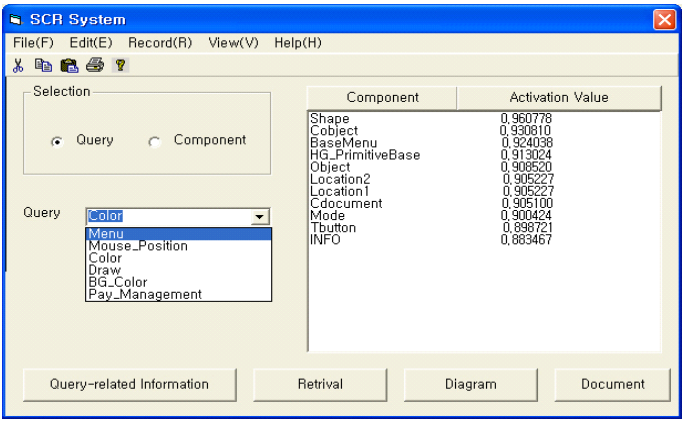


Fig. 3. Component Retrieval Result

the process of component input, retrieval is achieved by the addition of components to the components already existing in the database. The input interface identifies the relationship between the new components and the query. And in the representation of the relationship between the query and the component, by showing the permissible queries in the list, errors can be prevented. Fig. 3 shows the retrieval result of the components in order of the activation value with the input of the query Color. As a result of the component retrieval, 11 components are retrieved centering on the component Shape with the greatest number of reference. Components directly connected to Color are those directly connected up to the top five, Cobject and the rest are those indirectly connected. The retrieved components include the classes with the function of the query Color. The Document function for each of the components makes the access to the detailed function information of components possible. As a result, information may be obtained for the understanding of the retrieved components and the selection of components for reuse.

3 Performance Evaluation

3.1 E-SARM Efficiency

This paper has analyzed and saved the software components in terms of the information of the class, the relationship between classes, member function, and the diagram information and made it possible to reuse components directly on the diagram after retrieving them through E-SARM algorithm. For this, facet classification of the components has been achieved and the relationship with the query has been defined. Table 2 is a result of the simulation of computation of the activation value for the number of components. The simulation environment is represented in an arrangement of the size n by n with the column showing

Table 2. Number of Activation Value Computation

Component Size	SARM	E-SARM
200	22085.41	17108.90
400	48637.57	33964.71
600	82268.17	56627.29

Table 3. Average Number of Reference to Components

Component Size	SARM	E-SARM
200	6.3	4.75
400	7.88	5.48
600	11.2833	7.4

Table 4. Number of Components Retrieved

Query(10)	Retrieved relevant component	Retrieved unrelevant component	Unretrieved relevant component	Unretrieved unrelevant component
Random	11.5	1.4	1.3	24.5

the query, the row showing the component, and the initial value of each column and row showing the connection between the query and the component. The result shows that the retrieval time increases with the increase of components. This problem can be solved by removing the part with a small number of references from the algorithm. Table 3 shows the comparison of the numbers of the actual retrieval with the same standard as that shown in Table 2. This result shows that with the higher standard for removal the number of computation decreases and the number of reference of the component also declines. Comparing with the existing SARM algorithm after simulation, the average number of reference of components with E-SARM algorithm has been reduced by about 37.8%.

The result of retrieval of similar components for the query has been also examined to measure recall and precision of the retrieval system. Recall has been measured by the number of relevant components retrieved for the number of relevant components among the whole, and the precision has been measured by the number of relevant components retrieved for the number of the whole components retrieved. 10 queries were randomly selected, and the average number

Table 5. Retrieval Effectiveness

Classification	Average	
	Recall	Precision
E-SARM	89.8	89.1

of the retrieved components for each query was measured. Table 4 shows the average number of components retrieved for each of the queries. Table 5 shows the recall and the precision of these results. This experiment has shown that E-SARM has more efficiency because it maintains the retrieval result of SARM to the maximum with a high retrieval speed.

3.2 Component Reusability

As the components are retrieved in order of their activation value, components are examined to see if they are the wanted ones in order of the connectivity. The probability of selection of the wanted component is calculated by examining the components in order of their activation value for the retrieval. This means that the higher the activation value, the higher the probability of selection of the component, and thus easier to find the relevant component. Table 6 shows the cost of selecting the components with the retrieval method suggested by our study and those with pattern matching.

Table 6. Cost of Component Selection

Pattern Matching		E-SARM	
Cost	Selection Rate f(n)	Cost	Selection Rate p(n)
C_1	1/10	C_1	$P_1 = 0.325$
C_2	1/10	C_2	$P_2 = 0.25$
		...	
C_{10}	1/10	C_2	$P_{10} = 0.025$

As the pattern matching requires examining components one by one, the cost of the first component will be:

$$V_{PC1} = C_1 \cdot \frac{1}{10}$$

Consequently, the total cost(V_{PC}) of retrieval with pattern matching will be:

$$V_{PC} = C \cdot \frac{1}{10} + 2C \cdot \frac{1}{10} ... + 10C \cdot \frac{1}{10} = 5.5C \quad (1)$$

With E-SARM, the component retrieval cost for the first one with be:

$$V_{SC1} = C_1 \cdot P_1$$

And the total cost (VSC) of component retrieval will be:

$$V_{SC} = C \cdot P_1 + 2C \cdot P_2 + 3C \cdot P_3 ... + 10C \cdot P_{10} = 2.65C \quad (2)$$

This shows that retrieval costs with E-SARM are less than those with pattern matching. The reason is that E-SARM retrieves the components in order of the activation value, which means easier retrieval of reusable components. If we

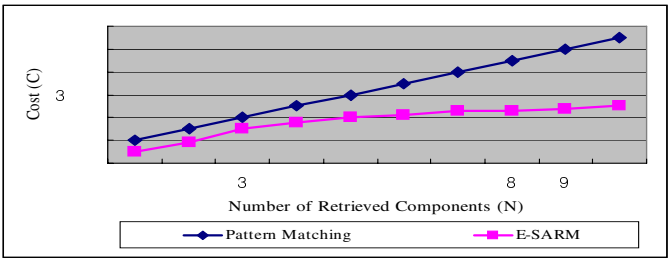


Fig. 4. Cost of Components Selection(Vc)

convert the formula into a general one using (1) and (2), the component selection cost with the pattern matching method will be:

$$V_{PC} = C \cdot \frac{1}{n} + 2C \cdot \frac{1}{n} ... + nC \cdot \frac{1}{n} = C \cdot \frac{(n+1)}{2} \tag{3}$$

The component selection cost with E-SARM will be:

$$V_{SC} = 1C \cdot P_1 + 2C \cdot P_2 + 3C \cdot P_3 ... + nC \cdot P_n \tag{4}$$

In the formula (3), the cost of n increases in proportion to n's increase; in the formula (4), as the probability P decreases as n increases, it gets a stable vale after some increase. Fig. 4 shows the two formulas in graph forms. It shows that the bigger the retrieved components (n), the more the cost of component selection reduces, and more so with E-SARM than with the pattern matching method. Consequently, the cost for component retrieval decreases with bigger components and E-SARM can retrieve the wanted component for reuse more rapidly as the retrieval is done in order of activation value.

4 Conclusion

This paper has improved SARM for retrieval of the object-oriented components so as to retrieve similar components, and established the object-oriented software component reuse system that allows the reuse of the components directly. Components are syntactically analyzed for their reusability and the interface is used to define the component-query relationship. Also, by removing the parts with a small number of references in the retrieval process, the time taken for retrieval in simulation has been reduced by 37.8% in comparison with SARM. As a result of simulation of E-SARM, it has been proved that though it has a disadvantage of slower speed than with the pat-tern matching, the retrieval for similar components is possible and the formula showed that the selection of components is more efficient as the result of retrieval comes out in order of the activation value related with the connectivity. Again, as the retrieval is performed in order of activation value, the cost for component selection is lower than with the pattern matching method. In summary, this paper has been confirmed to be a

more efficient system for reusing the object-oriented software component with a more efficient retrieval method in comparison with the existing reuse systems. In the future research, wider queries for diverse components and standardization are needed and an expansion method for showing the relationship between the analysis and the design procedure is required.

References

1. Heineman, G.T., Councill, W.T.: Component Based Software Engineering, pp. 143–160. Addision-wesley, London (2001)
2. Larsson, M., Crnkovic, I.: Configuration Management for Component-based System. In: Software Configuration(SCM-10), 23th ICSE Toronto, Canada (2001)
3. Washizaki, H., Yamamoto, Y., Fukazawa, Y.: A Mertics Suite for Measuring Reusability of Software Components. In: 9th IEEE International Symposium on Software Metrics, IEEE Computer Society Press, Los Alamitos (2003)
4. Henninger, S.: Information Access Tools for Software Reuse, System Software, 231–247 (1995)
5. Kim, G., Han, J., Song, Y.: Design Pattern Based Component Classification and Retrieval using E-SARM. The. Transactions of Korea Information Processing Society 11(5), 1133–1142 (2004)
6. Casanova, M., et al.: Supporting Evolution in Component-Based Development using Component Libraries, In: Seventh European Conference on Software Main-tenance and Reengineering(CSMR'03) (March 26-28, 2003)
7. Kon, F., Campbell, R.: Dependence Management in Component-Based Distributed Systems. IEEE Concurrency 8(1), 26–36 (2000)
8. Markus, G., Derek, B.: Case-Based Reuse of Software Examples. In: GWEM 2003 Proceedings of the Workshop on Experience Management (April 2003)
9. Ricardo, B.Y., Berthier, R.N.: Modern Information Retrieval. Addison-Wesley, Reading (2000)

The Clustering Algorithm of Design Pattern Using Object-Oriented Relationship

Gui-Jung Kim[1] and Jung-Soo Han[2]

[1] Department of Bio Medical Engineering, Konyang University,
Nae-Dong, Nonsan, Chungnam, Republic of Korea
gjkim@konyang.ac.kr
[2] Division of Information & Communication, Cheonan University,
Anseo-Dong, Cheonan, Chungnam, Republic of Korea
jshan@cheonan.ac.kr

Abstract. Clustering is representative method of components classification. But, previous clustering method that uses cohesion and coupling cannot be effective, because design pattern has consisted by relation between classes. In this paper, we classified design patterns with special quality of object-oriented relationship. Classification by clustering had expressed higher precision than classification by facet. Therefore, can do that it is effective that classify design patterns using clustering algorithms that is automatic classification method. When we are searching design patterns, using classification of design patterns we can com-pare and analyze similar patterns because similar patterns is saved to same category. Also we can manage repository efficiently because of using and storing link information of patterns.

1 Introduction

In order to use object-oriented methodology, we need practical concept and standardized pattern, and selection and application of the right objects in each context. It's design pattern that is standardized concretely for application to programming without stopping only by suggesting object oriented design concept theoretically[1][2][3]. Design pattern is a solution in realization course to maximize reusability and modularity, the biggest advantage of object-oriented methodology. There are hundreds of design patterns announced and known publicly by PLoP(Pattern Languages of Programs), design pattern conference in the U.S. and Euro Plop in Europe[4][5]. Efficient management of components is necessary to improve reusability of these increasing patterns[6]. Existing clustering is mostly done between classes or in classes, so this kind of clustering has used cohesion and coupling of class or module[7]. However, this way is not efficient to cluster design pattern focusing on relation and structure between classes. Therefore this study suggests pattern clustering algorithm for clustering design pattern consisted of relation between classes. Pattern clustering process takes two-course classification for pattern's quality. In the first course, pattern is classified by function and in the second course, by structure. Pattern algorithm is

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 997–1006, 2007.
© Springer-Verlag Berlin Heidelberg 2007

used in classifying with pattern structure. Suggested clustering of design pattern classifies pattern by pattern clustering algorithm and saves it using link information of pattern, for this helps us to manage repository efficiently and redesign system using pattern information.

2 Related Work

2.1 Gamma's Pattern Classification

Gamma classified design pattern into creational patter, structural pattern and behavioral pattern according to pattern's role. Creational pattern, a pattern to provide comprehensive solution for deciding creation method of objects, suggests method to organize and capsulate class definition and creation method of objects. Structural pattern, a pattern giving solution for method to compose class and objects in much bigger structure, provides general method to organize them when objects with different function play a role through cooperation. Behavioral patter, a pattern used to organize, manage and combine object's behaviors, is mostly used in algorithm performance like dividing function between objects [2].

2.2 Facet Classification System of GTE Corp

The system of GTE corp. proposed by Diaz is a representative reuse system using facet classification method [8]. This method, by which expansion of components, a weak point of classification method by enumeration was improved, demonstrates one component with several facets after compounding component's common quality and expressing one facet. This method expresses only foundation class of components, so classification is simple and easy to understand and expand[9][10]. Query can be corrected or newly asked using synonym management method for extracting components with similar function at the time of retrieval failure. However, there's a limit that it's fixed once classification is designed. It's difficult to detail their relationship and to deal synonyms when facet items are increasing. There's another weak point that retrieval time can be long.

2.3 Concept Analysis for Pattern Extraction

Concept analysis makes it possible to group objects with common attributes. In application of concept analysis, object is a group of class and attribute is relation between classes. The first view of concept analysis is context - collection of objects, collection of attributes and binary relation between object to determine attribute of each object and attribute[11]. Each relation links binary relation of class couple for application here. Concept is collection of objects with common attributes. Namely, it's a group of all objects sharing attributes. In general, concept is made by (X, Y).

$$X = \{o \in O \mid \forall a \in Y; (o, a) \in P\} \tag{1}$$

$$Y = \{a \in A \mid \forall o \in X; (o, a) \in P\} \tag{2}$$

O is an object. A is attribute. P is binary relation. X is concept extent. Y is concept intent. The main point for using concept analysis is that design pattern is concept generally. Actually, concept is composed of class group sharing relation in common. This class group is acquired by pattern found in source code or design while the number of attributes determines complexity of pattern [12].

3 Pattern Clustering

3.1 Pattern Clustering Process

Pattern Clustering Process is taken with two stages in Fig.1. In the first stage, functional classification is done by pattern's role. In the second stage, structural classification is done using relation between classes of patterns. A user can form user pattern with class diagram of UML, and then save pattern in pattern library through cluster-ing process. In clustering using pattern structure, it's difficult to define as related pattern though a pattern structure has a part to correspond. Therefore, the stage of functional classification should be taken before clustering with pattern structure.

In functional classification stage, pattern is classified into three kinds according to pattern's role. The pattern giving a comprehensive method to decide creation way of objects is classified into creational pattern. The pattern showing

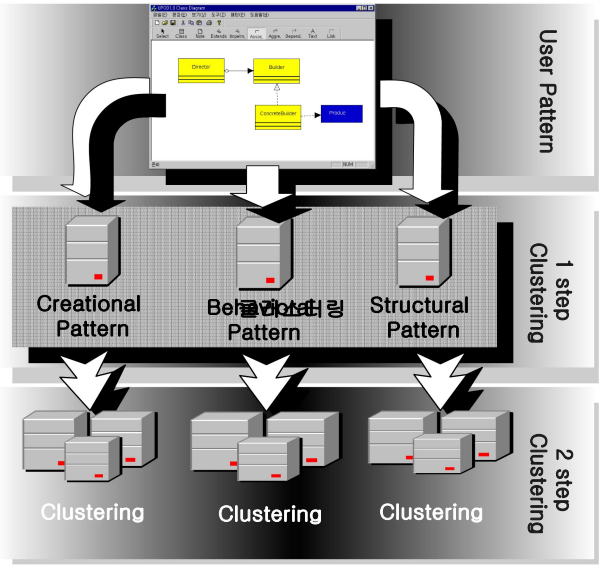


Fig. 1. Pattern Clustering

a general method to organize objects is classified into structural pattern. The pattern doing a role to classify function between objects is classified into behavioral pattern. In structural classification stage, clustering is done using pattern's structural form. We used clustering algorithm for clustering one of three foundation patterns of Gamma and pattern with the same structure among added pattern by a user to the same category.

3.2 Pattern Clustering Algorithm

Pattern clustering algorithm clusters to the same category when there's corresponding pattern with added pattern in foundation patterns. Pattern's structure is expressed with relation between classes in class diagram of UML[]. To compare patter's structure, one pattern is transformed into a group of order pair.

$$P = \{R_k(i,j) \mid (i,j) \in R, i \in C, j \in C, 1 \subseteq k \subseteq n\} \tag{3}$$

P is patter's order pair. R is relation of class. Proxy pattern in Fig.2 can be ex-pressed with order pair, P= G(2,1),S(3,2).

In each order pair, an alphabet letter means relation of class diagram and numbers mean two classes. <Table 1> shows abbreviated words to express relation of class diagram. An added pattern by a user in Fig. 3 can be expressed with order pair, P=S (1,2), G (1,3), G (4,3), S (5,4), G (6,5), G (7,5). An added pattern shows expanded proxy pattern. In this pattern, algorithm to locate proxy pattern is completed by comparing order pair of two patterns.

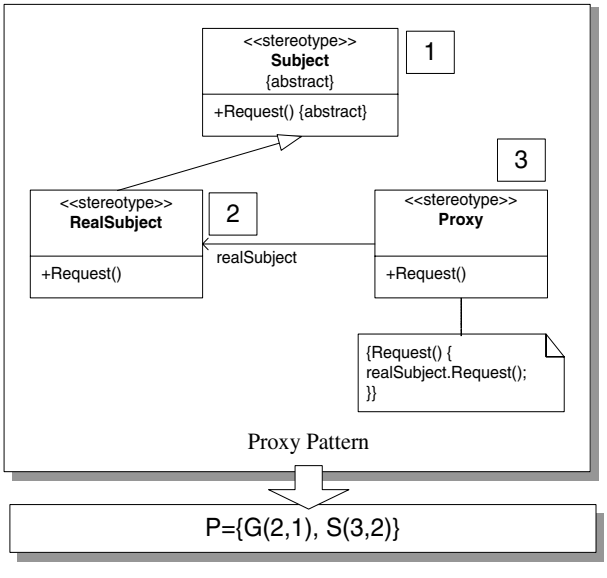


Fig. 2. Proxy pattern's order pair

Table 1. Class Relationship

Relationship	Symbol
Association	S
Generalization	G
Aggregation	E
Composition	C
Dependency	D
Realization	R

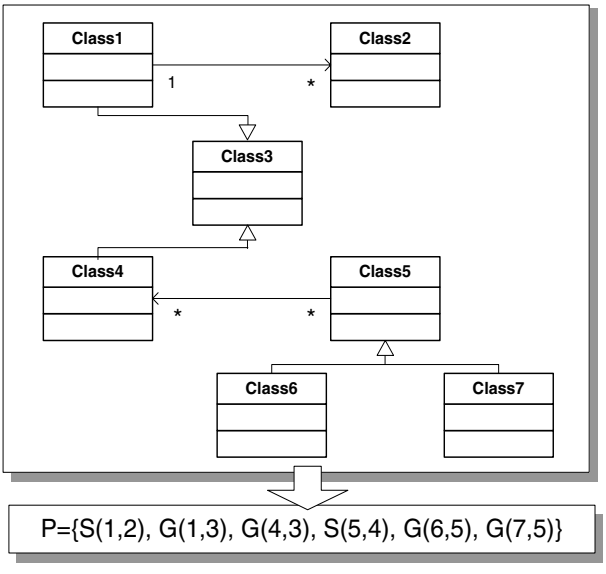


Fig. 3. Added pattern order pair

Initial Step $P_1 = \{R_1 = (i_1, j_1) \mid a \to i_1, b \to j_1\}$

Step $2 \subseteq k \subseteq n$

$$P_1 = \{R_k(i_k, j_k) \mid \text{if } i_k = i_1, \text{then } i_k \to a, j_i \to x$$
$$\text{if } j_k = i_1, \text{then } i_k \to x, j_i \to a$$
$$\text{if } j_k = j_1, \text{then } i_k \to x, j_i \to b$$
$$\text{if } i_k = j_1, \text{then } i_k \to b, j_i \to x$$

Fig. 4. Pattern order pair transform algorithm

In order to compare foundation pattern expressed by order pair with an added pat-tern; first, it's necessary to transform class expressed by numbers. Fig.4 shows pattern order pair transformation algorithm. Transformation should be done to prevent from bringing a different result without comparing each other whenever order pair changes because it's possible class number is optional, and compare regardless of changing order. The transformation process changes the first order pair into any letters. Here, the first class expressed in order pair is changed into a, and the second order pair is changed into b. In the rest order pair, the same number with one of the first class in the first order pair is changed into a, and the same number with one of the second class is changed into b. A Class coupling with class changed in order pair is expressed with x because the's no important meaning in comparison even though it is expressed with any number.

$$P = G(1,3), G(4,3), G(6,5), G(7,5), S(1,2), S(5,4)$$
$$\Rightarrow P = G(a,b), G(x,b), G(6,5), G(7,5), S(a,x), S(5,4)$$

Like above, class of the first order pair is transformed into (a,b), 1 into a and 3 into b. In other classes, 1 into a , 3 into b, and class coupling with order pair(a,b) into x because there's no relation in comparing with any class number.

In Fig.5, pattern comparison algorithm, Pf is foundation pattern, Pa is added pattern after transformation. In order to compare added pattern with foundation pattern, a user transforms foundation pattern and added pattern with transformation algorithm and then compares if these two patterns is same. To find a pattern related with foundation pattern among some order pair showing relation of added pattern, these order pairs are transformed in order. After this transformation , in order pair satisfies a conditional formula (4) below, added pattern,

```
//foundation pattern transformation
Pf = {(Pf1, Pf2, Pf3, ....., Pfk) | Transformation algorithms of foundation pattern}
// pattern order pair comparison
for (i = 0; i < n; i++)
{
    //additional pattern transformation for comparison
    Pf = {(Pa1, Pa2, Pa3, ....., Pak) | Transformation algorithms of addition pattern}
    //if addition pattern include foundation pattern, save and exit
    Pf ⊂ Pa : Break Addition();
    // if patterns structure Not same, change order pair
    Pa1 → Pan,   Pak → Pa(k-1)
}
```

Fig. 5. Pattern order comparison algorithm

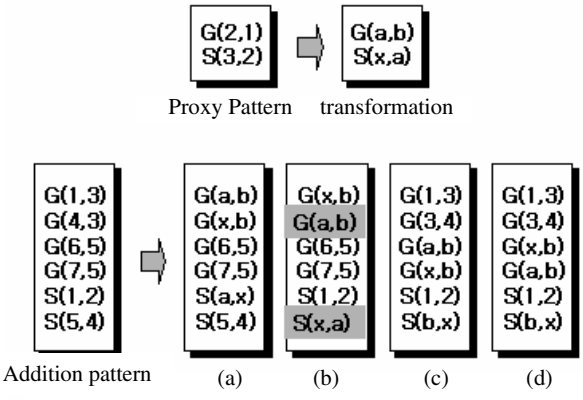


Fig. 6. Pattern comparison process

Pa is clustered into foundation pattern, Pf. If order pair doesn't satisfy a conditional formula (4) even after transformation of all order pair, added pattern is compared with other foundation pattern.

$$P_f \subset P_a \tag{4}$$

Fig.6 is a concrete diagramming of comparison process of proxy pattern and added pattern. Proxy pattern is composed of one association relation and one inheritance relation. Proxy pattern, G(1,3) is transformed to G(a,b) to search for association relation and inheritance relation of the same pattern among user patterns. In other order pair, 1 is transformed into a, 3 into b. After transformation, order pair of proxy pattern becomes G(a,b), S(x,b). A user pattern is compared through transformation process and has 4 association relations. Among these relations, to find the one with proxy pattern's structure, one of them will be compared with order pair transformed from proxy pattern through transformation process. If there are order pairs having structure of G(a,b),S(x,b), a user pattern is clustered to the group of proxy pattern. Otherwise, proxy pattern and added pattern can't be called the same structure pattern and compar-ing will be continued over again.

4 Performance

In this paper, we used 30 patterns announced at the PLOP(Pattern Languages of Programs) conference for an efficiency test [4][5]. We classified 30 patterns by method suggested in this paper and facet method. Facet method is a method to classify facet items after components' common qualities are made into several facet items. Precision of suggested method was measured checking relationship between classification result by clustering and real information of patterns.

Table 2. Pattern classification

index	pattern	# of class	Proposed category	classification	facet classification category	classification..
1	Order/Shipment Pattern	6	memento	o	memento	o
2	The Stock Manager analysis Pattern	8	prototype	o	prototype	o
3	Rule Object[PLOP2000]	13	mediator	o	mediator	o
4	The Object Filter and Access Control Framework	18	strategy	o	strategy	o
5	Legacy Wrapping[PLOP2000]	6	adapter	o	adapter	o
6	modifier[PLOP2000]	8	prototype	x	Factory method	x
7	Protocol System Architecture	15	bridge	o	proxy	x
8	Strategized Concurrency	5	state	o	state	o
9	Reductor[PLOP2000]	9	command	o	command	o
10	Phrasebook Pattern	4	builder	o	builder	o
11	Two Phase Commit[PLOP99]	7	decorator	o	decorator	o
12	Composite Transaction[PLOP99]	7	composite	o	composite	x
13	Matcher-Handler[PLOP99]	6	observer	o	observer	o
14	Autherntcator Pattern	5	prototype	x	Builder	x
15	Alternator[PLOP99]	6	state	o	state	o
16	Mayfly[PLOP99]	10	integrator	o	integrator	o
17	ACEE architecture[PLOP99]	5	observer	o	observer	o
18	Abstract Machine[PLOP99]	11	Abstract factory	o	Abstract factory	o
19	Grafcet Pattern[PLOP99]	5	command	x	integrator	o
20	Composite Calls[PLOP99]	10	composite	o	composite	o
21	Data Filter Architecture	9	proxy	o	proxy	o
22	Emissary Pattern[PLOP98]	5	strategy	o	state	x
23	Courier[PLOP98]	6	mediator	o	mediator	o
24	Override Current Processing	5	command	o	command	o
25	Substitution Pattern[PLOP98]	5	strategy	o	strategy	o
26	Delegation Pattern[PLOP98]	4	proxy	o	proxy	o
27	Reliable Hybrid Pattern	16	composite	o	composite	o
28	The Dynamic Template Pattern	6	observer	o	observer	o
29	Distributed Proxy[PLOP97]	9	proxy	o	proxy	o
30	Virtual Proxy[Larman98]	6	proxy	o	proxy	o

Table 3. Classified pattern precision

pattern	prposed category	facet classification category
precision	90%	83.33%

Table 4. The Number of class for pattern repository

Pattern / *repository*	10	20	30
existing system(# of class)	91	64	235
proposed system(# of class)	68	103	145

In <Table 2>, existence of relation was surveyed after classifying in two ways, suggested method and facet method. <Table 3> shows that precision is about 90% in category of suggested method and about 83% in facet method. This means that precision by facet method is lower than one by suggested method. In facet method, selection of facet items is optional according to pattern's quality, so there may be difference of precision according to choice of exact facet items. In pattern clustering classification, patterns are classified by forms expanded from

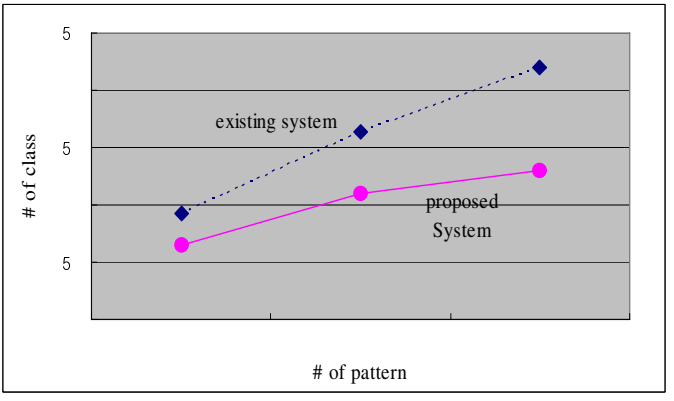


Fig. 7. Comparisons for pattern repository size

foundation pattern, for patterns in the same category can diminish the size of repository as repeating classes with only link information on foundation pattern when you try to save structures of expanded patterns. <Table 4> shows the number of classes saved in repository. Fig.7 is the size comparison of repository.

5 Conclusion

This paper proposes pattern classification using pattern's structure for efficient management of pattern. A pattern is classified into one among 23 categories by pattern clustering algorithm and then stored when pattern's structure and information is input. Pattern clustering algorithm is the method to use class structure of patters. By this method, added pattern is expressed with order pair and then clustered to one group if there's corresponding part with foundation pattern. Similar pattern retrieval is also possible by searching for patterns classified in category. Previous clustering method is not efficient, so we propose pattern clustering algorithm for design pattern classification. Classification by clustering expressed higher precision than classification by facet and saved information repository as the number of repeating classes with only link information at the time of storing patterns. From now on, the study of expanded clustering method for when added pattern by a user includes more than two foundation patterns is demanded. Research on various methods that make it possible for a user to add a foundation pattern when a pattern isn't classified to Gamma's foundation patterns should also be done.

References

1. http://www.omg.org
2. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Pattern: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1995)

3. Kingston, J.H., Lynn, B.Y.-S.: A Software Architecture for Timetable Construction. In: Burke, E., Erben, W. (eds.) PATAT 2000. LNCS, vol. 2079, pp. 342–352. Springer, Heidelberg (2001)
4. http://st-www.cs.uiuc.edu/~plop
5. http://www.argo.be/europlop
6. Kim, G.-J., Song, Y.-j.: Design Pattern Based Component Classification and Retrieval using E-SARM. The KIPS Transactions 11-D(5), 1133–1142 (2004)
7. Anquetil, N., Lethbridge, t.c.: Experiments with Clustering as a software Remodularization Method. In: Proceedings of the 6th Working Conference on Reverse Engineering, pp. 235–255 (1999)
8. Prieto-Diaz, R., Freeman, P.: Classifying Software for Reusability. IEEE Software 4(1), 6–16 (1987)
9. Kim, G.-J., Song, Y.-j.: Efficient pattern classification and retrieval supporting design pattern reuse. In: Proceeding of the IASTED International Symposia, pp. 511–517 (2001)
10. Kim, G.-J., Han, J.-S., Song, Y.-J.: Efficient Management of Pattern Information for Similar Design Pattern Retrieval. In: Proceeding of the SNPD 01 International Conference, pp. 444–449 (2001)
11. Grootjen, F.A., van der Weide, Th.P.: Conceptual Query Expansion, ICIS report NIII-R0406 (January 2004)
12. Tonella, P., Antoniol, G.: Object Oriented Design Pattern Inference. In: Proceedings of the IEEE International Conference on Software Maintenance, pp. 230–238 (1999)

Modeling Parametric Web Arc Weight Measurement

Wookey Lee[1], Seung-Kil Lim[2], and Taesoo Lim[3,*]

[1] Informatics Lab., Dept. I.E., Inha University, Incheon, Korea
{[2] Dept. of e-BusinessIT, [3] Dept. of Computer}, Sungkyul University,
Anyang-8 Dong, Manan-Gu, Anyang-city, Kyunggi-Do, Korea
Tel.:+82-31-467-8909; fax.: +82-31-467-8067
wook8174@yahoo.com, {seungkil, tshou}@sungkyul.edu

Abstract. An efficient searching for the Web contents become more important than ever before as the Web evolves and the number of users increases explosively. In this paper, an arc weight measure is focused to devise the relevance among Web objects with which a new method is presented for searching Web objects in a Web site. By this measure, an optimization model is derived to generate spanning trees embedded with traditional Web search mechanism. Case studies are performed with real Web sites and the corresponding results are considered in order to evaluate the possibility of using the new Web search mechanism.

1 Introduction

The World Wide Web is a collection of Web sites and its Web contents. The Web evolves continuously and changes dynamically since new Web sites are born and the old ones disappear simultaneously, and contents of those Web sites are updated periodically or automatically. While the Web contains vast amount of information and provides an access to it at any places and any times, that is a prize beyond our reach without efficient searching tools. Therefore, efficient searching for the Web contents become more important than ever before as the Web evolves and the number of users increases explosively. Portal sites with search engines are popular and commonly used tools for searching Web contents, although some promising efforts such as semantic Web have been continued for more efficient and effective use of the Web.

Most of portal sites have their search engines, which are used to find relevant Web contents for users' search queries. For efficient responses to users' queries, many portal sites have their index databases in which a collection of pointers to Web pages with occurrences of indexed words. Search engines find relevant Web contents by seeking indexed keywords (that are related to the query strings given by users) in the index databases. Portal sites update their index databases using special-purpose Web clients, called as Web spiders, search crawlers, or Web robots. Spiders send HTTP requests to a set of target Web sites to fetch Web pages of those sites. Since a Web site has its homepage and Web contents linked each other, search crawlers fetch the

* Corresponding author.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1007–1016, 2007.
© Springer-Verlag Berlin Heidelberg 2007

homepage of the Web site first and then obtain other Web contents by traversing hyperlinks within Web pages. Commonly used techniques to traversing referenced links are breadth first search and depth first search. Regardless of which techniques are used to traverse Web contents of a Web site, it is necessary to avoid traversing a Web content that has already been visited and fetched since cycles or circuits among links within Web pages may result in ineffective and/or inefficient collection of Web contents with search crawlers.

Since a Web site consists of a homepage (that can be accessed by its domain name) and many other Web contents linked each other (that can be located with its corresponding URLs or by clicking links within its Web pages), Web contents (including the homepage) of a Web site can be represented as a tree consisting of a set of nodes and associated arcs. Here, a Web page corresponds to a node, while links within the Web site among Web pages become directed arcs of the trees. If we transform Web contents of a Web site into a corresponding tree, we can find a set of paths through which any Web page of the Web site can be accessed from the homepage and all other pages within the Web site. This set of paths can be obtained from finding a spanning tree for the Web site. Since a spanning tree does not have any cycles or circuits among nodes, search crawlers can avoid revisiting Web contents that has already been visited to fetch the contents if the search crawlers send HTTP request messages according to paths obtained by the spanning tree.

There are works on exploiting the hierarchical structure of the Web. Statistical analysis shows the fact that people jumps to a randomly selected top-level page of a site, which indicates that there is a strong human tendency for information to be organized hierarchically [4, 14]. Additionally, users of Web information, accessible over the Web, require assistance by appropriate visualization methods [16] in order not to be lost in hyperspace [11]. The hierarchical visualization of the Web has provided many benefits such as enhancing spatial context, easing jumps to specified URLs, reducing disorientation, providing a sense of the extent of a particular Web site without getting any details, and acting as a visual surrogate for the users [7, 12, 16]. In performance perspective, the hierarchical "data structure" can support fast search for the Web searcher, e.g., Web robots that they should crawl on the fly under limited resource constraints.

We use PageRank to measure relative importance of a Web page to other Web contents linked to the Web page with a damping factor of 0.85 that is used by Google. Note that PageRank is a link weight that can be interpreted as the degree of relative importance of a Web content to other Web contents linked directly to the Web content in a Web site. For more details of PageRank, see [13]. In this paper, we devise a new arc measurement method using PageRank and develop a binary integer programming model.

This paper is organized as follows. After discussing the characteristics of Web site graph, we describe arc weighting methods and a formulation to find the optimal spanning tree. Next section gives results of case studies, and finally we summarize research results with some concluding remarks.

2 Web-as-a-Graph Approach

The WWW can be viewed as digraph with Web nodes and arcs, where the Web nodes correspond to HTML files having page contents and the arcs correspond to hypertext links interconnected with the Web pages. The Web-as-a-graph approach can be a starting point to generate a structure of the WWW that can be used for Web site designers, search engines, Web crawling crawlers, and Web marketers and analysts [1, 5]. Formally the Web directed graph $G = (N, A)$ can be represented with an arc function $x_{ij} : N^k \quad \{0, 1\}, \quad i, j \in N$ consists of a finite Web node set N, a finite Web arc set A of ordered pairs of Web nodes, and the Web arc elements (i, j) respectively, where $i, j \in N = \{0, 1, 2, 3, \ldots, n\text{-}1\}$, and $n = |N|$ the cardinality of Web pages. There is a mapping for the nodes corresponding to Web pages and the arcs to Uniform Resource Identifiers.

The Web-as-a-graph approach, however, has weaknesses such as unreachable paths, circuits, and repetitive cycles. The unreachable path is that a Web page sometimes can not be accessed from the usual path or a hub mode. The circuit as a cycle is that a client visits the same page again and again periodically. In order to generate a Web structure, the circuits and repetitive cycles should be detected and removed, without which a Web client may be lost in Cyber space through the complex cycles or may inevitably gather information with swallow depths from the root node. The Web-as-a-tree approach has several benefits in comparison with graph based model. One is that a Web site consists of a home page (e.g., default.html) that can be the root node from which a hierarchical structure can correspondingly be derived. The other is that the hierarchical structure can conceive very simple data structure so that the crawling performance to search the Web on the fly can be highly enhanced.

The typical hierarchical examples are breadth first search and depth first search with which Web catalogues, site maps, and usage mining can be pertained [8, 9]. The breadth first approach, including *backlink* count method [2, 9], has some advantages so that an 'important' Web page can be accessed within relatively fewer steps from its root node. It can statistically minimize the total number of depths. In the Web environment, however, the tree by breadth first approach may result extremely flat so that all the pages stick to the root node. On the other hand, the depth first approach is popularly adopted for practitioners to exploit a structure with a stack data format [6]. In structuring the Web, the approach is not appropriate because it may result in a long series of Web pages. It means that the series of pages entail mouse clicks, so that as much time consumption to access each page is required. The worst thing on the derived structures by these two approaches is that no measures or values, even no semantics can be endowed.

3 Structuring the Web

In this section, we propose two methods for determining arc weights and formulate an integer programming model for Web site structuring.

3.1 Arc Weights Measurements

In this research, we develop a new method for determining the arc weights that based on PageRank algorithm, well-known Web objects ranking algorithm. In PageRank equation (1), the node rank $R(i)$ of node i is derived recursively by Markov transition matrix a_{ij} that should be nonnegative and irreducible with primitive transition as well as aperiodic [10]. In equation (1), n is the number of nodes, and $InA(i)$ is the set of incoming arcs into node i. Google usually set the value of the damping factor d to 0.15 by which the vector converges either slowly or quickly with respect to the magnitude.

$$R(i) = (1-d) \times \left[\frac{1}{n}\right]_{n\times 1} + d \times \sum_{\substack{(i,j)\in A \\ j\in InA(i)}} a_{ij} R(j) \tag{1}$$

Even though the PageRank utilizes the arc structures, it focuses on “page rank” not on “arc rank.” Without an arc weight, structuring the Web can be a trivial problem. For example, if the nodes only have values, then any structures (e.g., a tree or the complete graph) with the node values results in the same total weight sum. In order to generate a significant structure from the Web, an arc oriented weight is needed.

In the PageRank, the weight of an arc gets the probability that a user visits from a Web node to another Web node. From the arc standpoint, the importance of a Web node is uniformly distributed to the outgoing arcs to the linked Web pages. Therefore, an arc weight w_{ij} from Web node i to an adjacent node j ($i, j \in N$) can be described as following equation (2), where $O(i)$ is the number of outgoing arcs from node i.

$$w_{ij} = \begin{cases} 1/n, \text{if } O(i) = 0 \\ R(i)/O(i), \text{otherwise} \end{cases} \tag{2}$$

However, this kind of arc weight of equation (2) has some limitations. One is that the arc weight needs to be changed for the saturated (leaf) arc. In PageRank, a node with no incoming arc is given the lowest value, sometimes equaling to 0. In that case, a decay factor (the minimum constant value) can be given. We know that if the saturated node and arc are deleted, then the weights of the others nodes will be altered. This means that the arc weight may not be properly evaluated. Another one is that the assumption of uniform distribution for the PageRank needs to be changed. The hypertext link can be a "move" from a node to another. From the arc weight perspective, difference between bigger weight node and lesser weight nodes is not defined in PageRank. The two moves to the node with less weight and to the node with more weight should be considered differently.

One idea for changing the arc weight is the notion of Euclidean distance such that both sides of the nodes can be taken into consideration. Wookey and Geller [17] used that weight like equation (3), where $Weight(i)$ can be a distance weight (e.g., $R(i)$). Using the Euclidean distance weight, the problem of saturated node can be resolved but cannot consider the directionality of arcs. In the cyclic instances, the equation (3) is not properly represented. The arc weights are considered to be equal although the node weights are different. Euclidean weight between the two nodes (i.e., average

between the two nodes) is not enough. Instead of simple weight, a harmonious average or k-mean measure, or semantic distance measures can limitedly be applied [3, 8, 9].

$$w_{ij} = \frac{Weight(i) + Weight(j)}{2} \tag{3}$$

We propose another alternative method to measure the arc weight by using parametric arc weight between the two adjacent nodes as follows, where the arc from node i to node j is a weighted average weight of the two adjacent nodes in terms of the parameter p, where $0 \le p \le 1$.

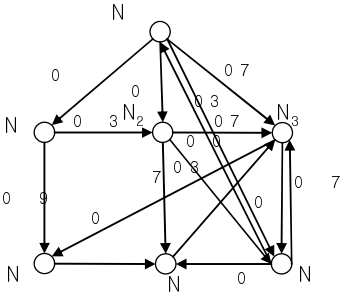
$$w_{ij} = p * Weight(i) + (1-p) * Weight(j) \tag{4}$$

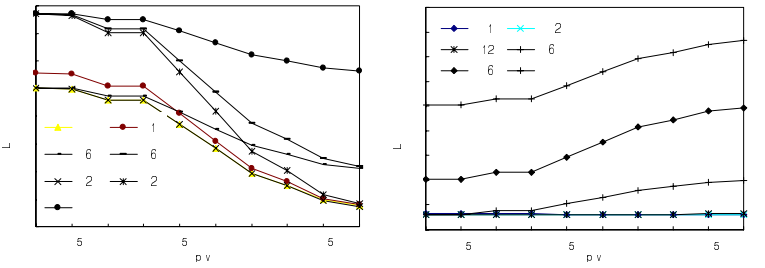
If the PageRank is applied, then the equation (4) will be as follows.

$$w_{ij} = p * \frac{R(i)}{O(i)} + (1-p) * \frac{R(j)}{O(j)} \tag{5}$$

It is an integrated representation on the arc weight such that various weights can be incorporated. The above equations give us an insight as a balanced weight with parameter p. The p is a parameter as an "indicator" to push from node i to node j and 1-p is to pull vice versa.

- When p = 1, it results in the pure "push method," that is the same as equation (2).
- When p = 0, it is pure "pull method."
- When p = 0.5, it is equal to that of equation (3).


N
N
N
N
N
N
N
N
N
0
0
0



(a) Decreasing link weights (b) Increasing link weights

Fig. 2. Monotonic characteristics of link weights

3.2 Formulation

Here, we formulate a binary integer linear program using the notation given below. We assume that all Web objects and hyperlinks among them are converted into corresponding directed graph. Also, the directed graph has its root node as one Web object that are selected by a user at a search engine from the list of URLs as explained in previous section.

- i, j indices for nodes that represent Web objects existing in a Web site, $i = j = 0, 1, \ldots, n$ (Here, node zero is the root node)
- t index for terminal nodes that are all other nodes except the root node, $t = 1, 2, \ldots, n$
- N set of all nodes
- A set of all existing arcs
- w_{ij} weight of arc (i, j) that represents the degree of relevance between two nodes
- x_{ij} equals 1 if arc (i, j) is selected to make rooted spanning tree, and 0 otherwise.

Now, we present a binary integer linear programming for finding the best tree structure maximizing the summation of arc weights.

$$\max \sum_{\substack{i,j \in N,\ j \geq 1 \\ (i,j) \in A}} w_{ij} x_{ij} \tag{6}$$

$$\text{subject to} \sum_{\substack{i \in N \\ (i,0) \in A}} x_{i0} = 0 \tag{7}$$

$$\sum_{\substack{i \in N \\ (i,j) \in A}} x_{ij} = 1 \qquad \text{for all } j \neq 0 \tag{8}$$

$$x_{ii} = 0 \qquad \text{for all } i \tag{9}$$

$$x_{ij_1} + \begin{cases} x_{j_m i} \le m & \text{if } m = 1 \\ \sum_{k=1}^{m-1} x_{j_k j_{k+1}} + x_{j_m i} \le m & \text{if } 2 \le m \le |N| - 1 \end{cases} \tag{10}$$

For all directed circuits starting from any node i, and returning to the same node through m intermediate nodes, where a directed circuit can be represented as an ordered list of nodes, $(i, j_1, j_2, ..., j_k, j_{k+1}, ..., j_{m-1}, j_m, i)$

$$x_{ij} \in \{0,1\} \qquad \text{for all } i, j \tag{11}$$

The web site consists of seven web pages including the root page that are represented as web page zero. Arcs among web pages are represented by arrows in which directions of the arrows means directions of arcs among the web pages. As a result, bidirectional arrows between two web pages indicate that one web page has a arc to the other web page each other. Numbers on the arrows are weights of the corresponding arcs. The function (6), which is to be maximized, denotes the sum of the weights for arcs that determine the spanning tree. Constraints (7) ensure that any of arcs toward the root page should not be included for making feasible spanning tree, while constraints (8) ensure that each of all terminal web pages should have only one web page that is arced directly to the terminal web pages. Constraints (9) and (10) eliminate all cycles existing in given web site. Constraints (11) specify that decision variables should be binary integers. The web site consists of seven web pages including the root page that are represented as web page zero. Arcs among web pages are represented by arrows in which directions of the arrows means directions of arcs among the web pages. As a result, bidirectional arrows between two web pages indicate that one web page has a arc to the other web page each other. Numbers on the arrows are weights of the corresponding arcs.

4 Case Study

We performed a case study to evaluate how to find structured search trees (that is, a rooted spanning tree) for given a Web object.

We test our approach of generating rooted spanning trees with real case. In this test, we assume a user selects a URL of a Web site from a search engine and the selected Web object (that is selected URL) becomes the root node of a directed graph for the Web site. Assume that selected URL is the homepage for a Web site of Samsung Electronics, www.samsung.co.kr. The number of only static Web objects in Samsung's Web site is 31 and number of hyperlinks are 88. We convert these Web objects and hyperlinks into corresponding directed graph using a prototype system, called as AnchorWoman. Fig. 3 shows information on nodes (Web objects) and arcs (hyperlinks among nodes) obtained with the prototype system. Fig. 4 gives the converted directed graph for Samsung's Web site, and an optimal rooted spanning tree is given in Fig. 5.

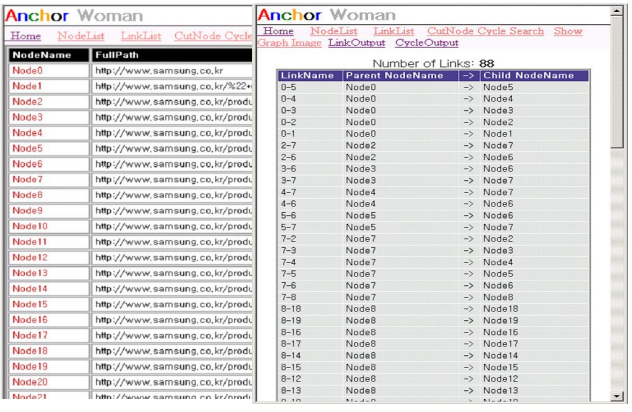


Fig. 3. Nodes and arcs information on www.samsung.co.kr

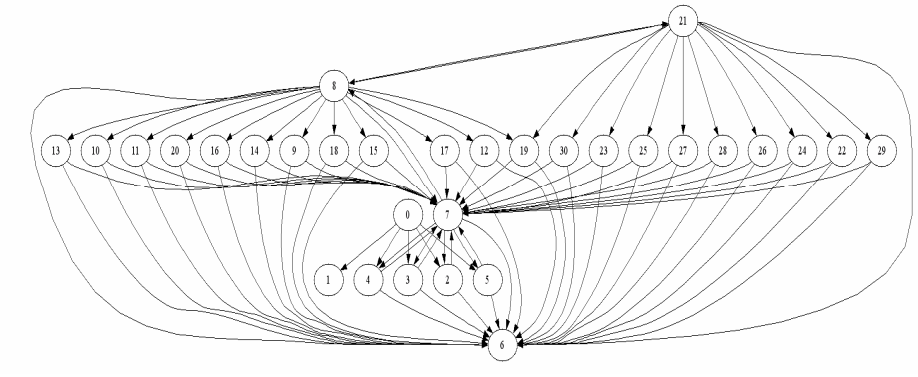


Fig. 4. Directed graph for www.samsung.co.kr

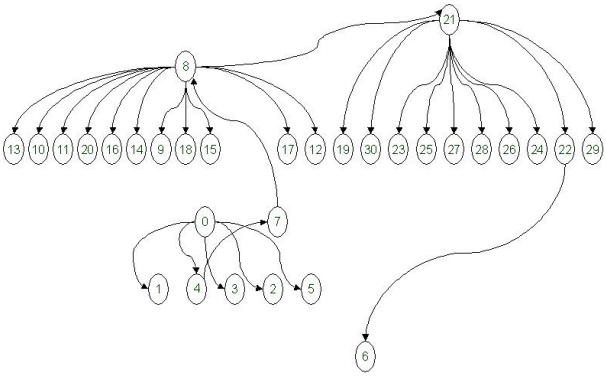


Fig. 5. Optimal hierarchical form of the example site

Although we do not summarize results, we performed similar computational tests for several distinguished electronic companies' Web sites. Included are Hitachi, Intel, Microsoft, NEC, and Nokia. Their Web sites have a number of thousands of Web objects and hyperlinks. The two binary integer linear programming models for these Web sites are solved within a few seconds with commercial optimization software CPLEX on a personal computer.

5 Conclusions and Further Researches

Parametric web arc weight measurements are suggested in this research. For the structured web search, we developed an optimization model to find the best rooted directed spanning trees. The interactive arc weighting methods do not use pre-computed node ranks, the initial values of node or arc ranks are required to initiate the interactive and iterative rank calculation. In the parametric web arc weight measurements, arcs are measured based on the degree of contribution for both nodes as a connectivity weight. The method can be adopted to incorporated node weights, and therefore generates consistent solutions with conventional node based as well as arc based measurements.

This research can be extended to several directions. First, there may be some cases that it is impossible to find a feasible rooted spanning tree when a user selects a URL from the list given by a search engine with the current structure of a Web site (that is hyperlinks among Web objects) although we can make a rooted spanning tree for any given root if a directed graph for a Web site satisfies strongly coupled condition. In this case, we can make a feasible rooted spanning tree by adding temporal hyperlinks to current structure of Web site. Web server of the Web site may perform this addition of temporal hyperlinks e.g., robots.txt. Also, we need to develop extended mathematical models for identifying temporal hyperlinks. We need to evaluate the effectiveness of the structured Web search suggested in this research in actual Web searching environments. Comparisons between traditional Web search and the structured Web search should be made for the purpose.

Acknowledgement

This research was supported by the Ministry of Information and Communication, Korea, under the College Information Technology Research Center Support Program, grant number IITA-2006-C1090-0603-0031.

References

1. Berry, M., Browne, M.: Understanding Search Engines: Mathematical Modeling and Text Retrieval, SIAM (1999)
2. Cooley, R.: The Use of Web Structure and Content to Identify Subjectively Interesting Web Usage Patterns. ACM Internet Technology 3(2), 93–116 (2003)
3. Demaine, E., Lopez-Ortiz, A.: A Linear Lower Bound on Index Size for Text Retrieval. Journal of Algorithms 48(1), 2–15 (2003)

4. Eiron, N., McCurley, K., Tomlin, J.: Ranking the web frontier. WWW, pp. 309–318 (2004)
5. Etzioni, O., Cafarella, M., Downey, D., Popescu, A., Shaked, T., Soderland, S., Weld, D., Yates, A.: Methods for Domain-Independent Information Extraction from the Web: An Experimental Comparison. In: Proc. AAAI, pp. 391–398 (2004)
6. Nivasch, G.: Cycle detection using a stack. Information Processing Letters 90(3), 135–140 (2004)
7. Getoor, L., Friedman, N., Koller, D., Taskar, B.: Learning probabilistic models of link structure. The Journal of Machine Learning Research 3, 1–29 (2003)
8. Lee, W.: Hierarchical Web Structuring from the Web as a Graph Approach with Repetitive Cycle Proof. In: Shen, H.T., Li, J., Li, M., Ni, J., Wang, W. (eds.) APWeb 2006. LNCS, vol. 3842, pp. 1004–1011. Springer, Heidelberg (2006)
9. Henzinger, M.R., Heydon, A., Mitzenmacher, M., Najork, M.: On near-uniform URL sampling. Computer Networks 33(1), 295–308 (2000)
10. Hou, J., Zhang, Y.: Effective Finding Relevant Web Pages from Linkage Information. IEEE TKDE 15(4), 940–951 (2003)
11. Lau, T., Etzione, O., Weld, D.S.: Privacy Interfaces for Information Management. CACM 42(10), 89–94 (1999)
12. Najork, M., Wiener, J.: Breadth-first Crawling Yields High-quality Pages. In: Proc. WWW, pp. 114–118 (2001)
13. Pandurangan, G., Raghavan, P., Upfal, E.: Using PageRank to Characterize Web Structure. In: Ibarra, O.H., Zhang, L. (eds.) COCOON 2002. LNCS, vol. 2387, pp. 330–339. Springer, Heidelberg (2002)
14. Pokorny, J.: Web Searching and Information Retrieval, Computing in Science & Engineering, pp. 43–48 (2004)
15. Tarjan, R.: Enumeration of the Elementary Circuits of a Directed Graph. SIAM Journal on Computing 2(3), 211–216 (1973)
16. Toyota, M., Kitsuregawa, M.: A system for Visualizing and Analyzing the Evolution of the Web with a Time Series of Graphs. In: Proc. HT, pp. 151–160 (2005)
17. Wookey, L., Geller, J.: Semantic Hierarchical Abstraction of Web Site Structures for Web Searchers. Journal of Research and Practice in Information Technology 36(1), 71–82 (2004)

Performance Analysis of EPC Class-1 Generation-2 RFID Anti-collision Protocol

Jeong Geun Kim, Woo Jin Shin, and Ji Ho Yoo

Department of Electronics and Information,
Kyung Hee University, Suwon, Republic of Korea
jg_kim@khu.ac.kr

Abstract. Radio Frequency Identification (RFID) has been considered an enabling technology for fast and reliable identification of massive objects. However simultaneous identification of multiple objects attached with electronic tag is challenging since responses from tags without coordination may collide, thereby resulting in a waste of radio resources. In this work we analyze the performance of anti-collision protocol designed for Electronic Product Code (EPC) Class-1 Generation-2 RFID systems and propose an operation algorithm for the anti-collision scheme. By using the information on collision statistics in the previous frame, our algorithm estimates the number of contending tags and allocates accordingly an appropriate number of slots in the next frame. Probabilistic analytic models are developed to calculate the time required to read a complete set of tags. Simulations are used to evaluate the performance of the proposed operation algorithm and to verify the adequacy of the analysis.

1 Introduction

A Radio Frequency Identification (RFID) system consists of multiple radio frequency (RF) tags and readers (also referred to as interrogator) [1][5]. The reader attempts to retrieve unique identification number by supplying the energy source with passive tags. The data transmission from tags to reader is done by scattering the electromagnetic wave back to the reader. Depending on the data bit, the tags modulate the carrier with two different frequencies while scattering. A collision may occur when multiple tags respond simultaneously to the reader's inquiry. Collision incurs a mixture of scattered waves and prohibits correct identification. To cope with this problem a certain form of coordination protocol is required. Such a technical procedure in RFID systems that facilitate the handling of collision without any interference is called *anti-collision* protocol [1].

As pointed in [5] the anti-collision problem in RFID systems is characterized by the following:

- A large number of tags may be involved in the arbitration process.
- The number of tags to identify varies over time.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1017–1026, 2007.
© Springer-Verlag Berlin Heidelberg 2007

- A tag never initiates communications without being asked by the reader.
- The capability of a tag is severely limited by practical restrictions on size, power, and cost.
- Asymmetric link quality between reader and tags.

Due to random nature of contention process and restrictions in processing capability, a highly effective and efficient approach to collision resolution is required in RFID systems.

Different anti-collision protocols have been proposed and analyzed in the literature [2][5][3][4]. In [3], Vogt proposed an adaptive technique for multiple tag identification in framed slotted ALOHA anti-collision system. In this scheme the number of active (or contending) tags is estimated and the system parameters are adjusted accordingly. However the tag selection procedure is different from the EPC standard and analysis for dynamic slot allocation was not included in the work. The authors in [2] proposed a memoryless protocol called Query-Tree (QT) for the tag identification problem. The merit of this protocol is memoryless, thus the current response of the each tag only depends on the current query of the reader. Theoretical results in both time and communication complexities are derived to evaluate the efficiency of the protocols. The authors in [5] analyzed a family of tree search algorithms in terms of arbitration time and consumed power. They also developed expressions for the decomposition of time slots into three basic types: zero, one, and multiple tag replies.

Among several anti-collision schemes our study focuses on EPCglobal's recent proposal that is basically based on framed slotted ALOHA [9]. In this protocol, the reader first selects a group of contending tags. Identification procedure then goes on as the reader sends a series of command queries to the selected group in an interactive manner. Since statistics of collision is completely determined by the number of contending tags and their choices of time slots, the reader must be capable of carefully controlling the number of contending tags and the number of slots in each frame. In order to determine optimum values for those key parameters, we first establish analytical models for two alternative operation principles: static slot allocation and dynamic slot allocation. Static allocation fixes the number of slots in the frame, whereas dynamic allocation adjusts it depending on contention level. For both principles, we develop the distribution of *identification time*, defined as the time taken to read out all the ID information from a complete set of tags. In order to verify the adequacy of the analytic models and investigate the performance of EPC anti-collision standard, we carry out extensive simulations using a discrete-event simulator.

The rest of the paper is organized as follows. In Section 2 we give a brief overview of EPC anti-collision protocol standard and describe the proposed operation algorithm. Probabilistic analytic models for static and dynamic slot allocation are also developed in the section. Numerical results and simulations are reported in Section 3, followed by concluding remarks in Section 4.

2 Anti-collision Protocol

2.1 EPC Class-1 Gen-2 Anti-collision Protocol

The EPC standard employs a variant of slotted ALOHA, known as framed slotted ALOHA, for anti-collision protocol. In this scheme, time is divided into a certain number of slots that constitute a single frame. Upon receiving the signal that indicates the beginning of identification process, each tag chooses a slot independently and transmits the data over the slot. Due to the nature of random access, a collision occurs when multiple tags select the same slot. Collision incurs a mixture of scattered waves and prohibits the reader from correctly retrieving the information.

Although elaboration on the detailed interactions between reader and tags is skipped, overall identification procedure can be summarized by the following signals:

1. **Select** (Reader → Tags): Select a group of tags to be involved in contention for the current frame
2. **Query** (Reader → Tags): Broadcast the number of allocated slots in the frame
3. **Reply** (Tags → Reader): Transmit data in a chosen slot
4. **Ack** (Reader → Tags): Indicate correct reception of information

Figure 1 shows how the identification procedure is performed in the EPC standard [9]. At the beginning of every round (or frame), the reader first sends the **Select** command to the tags indicating the matching bits and the address of bits to match on tag's side. Upon receiving this message, each tag determines whether to join the contention process in current frame by comparing the indicated bits in its memory block with the one in the **Select** message. Tag joins the contention process if both bit strings match. The **Query** command following the **Select** informs tags of the number of slots allocated in the frame. Each tag then independently chooses a random slot and sends a 16-bit random number. Unless

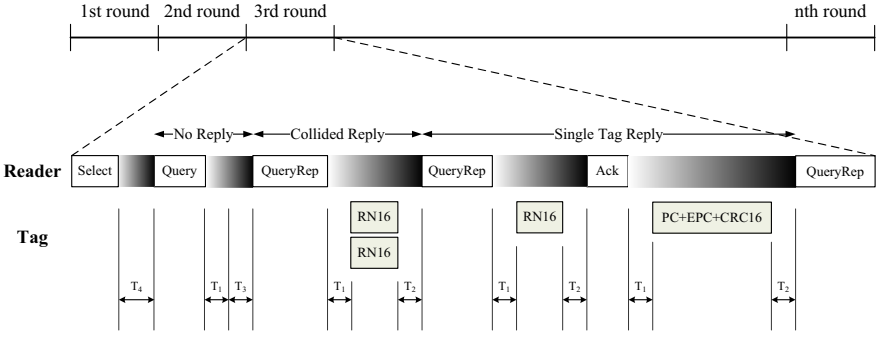


Fig. 1. Identification procedure in the EPC Class-1 Gen-2 system

collision occurs the reader correctly retrieves the information and sends the **Ack** message to the tag. The tag receiving the acknowledgement then responds with EPC code. If collision occurs the reader remains silent. Each slot in the frame is separated by the **QueryRep** command.

Throughput of the EPC anti-collision protocol, defined as the number of tags identified per second, absolutely depends on the parameters including (1) the number of selection bits, (2) the number of slots allocated in the frame. By configuring these at optimum values we can achieve optimal throughput, or equivalently the minimum *identification time* defined as the time taken to read out all the ID information from a complete set of tags.

2.2 Analysis

In this section we establish analytic models for the alternative operating principles on controlling the aforementioned parameters:

i. Static allocation: Fixed number of slots is allocated in every frame
ii. Dynamic allocation: The number of slots in the frame are dynamically allocated depending on contention level.

Notation
The following notations are used throughout the rest of this paper.

- N_t: the total number of tags
- N_s: the number of slots allocated in the frame
- O_n: the number of slots allocated in at the nth round
- U_n: the number of unidentified tags at the nth round
- V_n: the number of tags involved in the contention at the nth round
- S_n: the number of successfully identified tags only at the nth round
- C_n: the number of slots in which collision occurs at the nth round
- A_n: Elapsed time from the beginning in unit of slots by the nth round
- p_s: Probability that a tag is selected for contention with the selection bit size of s bits

$$p_s = \frac{1}{2^s}$$

Static Slot Allocation. Let $P_n(k)$ denote the probability that k tags remain unidentified after nth round, that is,

$$P_n(k) = P[U_n = k], \quad 0 \le k \le N_t.$$

The $P_n(k)$ is given by

$$\begin{aligned} P_n(k) &= \sum_{l=k}^{N_t} P[U_n = k \mid U_{n-1} = l] P_{n-1}(l) \\ &= \sum_{l=k}^{N_t} P[S_n = l-k \mid U_{n-1} = l] P_{n-1}(l) \qquad (1) \\ &= \sum_{l=k}^{N_t} \sum_{i=0}^{l} P[S_n = l-k \mid U_{n-1} = l, V_n = i] \binom{l}{i} p_s^i (1-p_s)^{l-i} P_{n-1}(l) \end{aligned}$$

with initial conditions

$$P_0(k) = \begin{cases} 0, & 0 \le k \le N_t - 1 \\ 1, & k = N_t \end{cases} \tag{2}$$

and the probability that u tags are successfully identified given w tags are involved in the contention for a fixed number of slots N_s is given by

$$P[S_n = u \mid U_{n-1} = v, V_n = w] = \frac{\binom{N_s}{u} \prod_{i=0}^{u-1} (w-i)((N_s-u)^{w-u} - H(N_s-u, w-u))}{N_s^w} \tag{3}$$

In the above equation, $H(n,m)$ denote the number of ways that at leats one slot has a single tag given n slots and m tags, that is given by

$$H(n,m) = \sum_{k=1}^{\min(m,n)} (-1)^{k+1} \prod_{j=0}^{k-1} (m-j)(n-j)(n-k)^{m-k} \frac{1}{k!} \tag{4}$$

$P_n(0)$ represents the probability that the number of unidentified tags is zero at nth round. In Section 3 we use this metric to represent *identification time*. $P_n(0)$ is calculated using (1) by recursive substitution with initial conditions (2).

Dynamic Slot Allocation. Estimation on the number of contending tags is obtained through observation on the contention results of the previous round. As in [3], a simple estimator can be given by

$$O_n \simeq S_{n-1} + 2C_{n-1} \tag{5}$$

We use this estimator in the following analysis.

The joint probability that u slots are accumulated by the nth round, v unidentified tags remain, w slots retrieve the data successfully, and x slots has a collision, is given by

$$P[A_n = u, U_n = v, S_n = w, C_n = x] = \sum_{p,q,r,s} P[A_n = u, U_n = v, S_n = w, C_n = x \mid A_{n-1} = p, U_{n-1} = q, S_{n-1} = r, C_{n-1} = s] \, P[A_{n-1} = p, U_{n-1} = q, S_{n-1} = r, C_{n-1} = s]$$

with initial conditions

$$P[A_0 = p, U_0 = q, S_0 = r, C_0 = s] = \begin{cases} 1, & p = 0, q = N_t, r = S, s = 0 \\ 0, & \text{elsewhere} \end{cases}$$

The above probabilities has non-zero values for the following conditions

$$\begin{aligned} p &= u - r - 2s \\ q &= v + w \\ w + x &\le r + 2s \end{aligned} \tag{6}$$

The conditional probability is equivalent to the following equation:

$$P[A_n = u, U_n = v, S_n = w, C_n = x \mid A_{n-1} = p, U_{n-1} = q, S_{n-1} = r, C_{n-1} = s] = \sum_{i=w+2x}^{q} \binom{q}{i} p_s^i (1-p_s)^{q-i} P[S = w, C = x \mid N_t = i, N_s = r+2s] \quad (7)$$

The conditional probability on RHS is the probability that j successful reading, k collisions, l empty slots, given n slots and m tags, that is given by

$$P[S = j, C = k, E = l \mid N_t = m, N_s = n] = \frac{j!}{n^m} \binom{n}{j,k} \binom{m}{j} (k^{m-j} - G_0(k, m-j) - G_1(k, m-j)) \quad (8)$$

where $n = j + k + l$, and G_1 denote the number of ways that at least one slot has a single tag, but does not have empty slot, given n slots and m tags,

$$G_1(n,m) = \sum_{k=1}^{\min(m,n)} (-1)^{k+1} \prod_{j=0}^{k-1} (m-j)(n-j) \left((n-k)^{m-k} - G_0(n-k, m-k)\right) \frac{1}{k!} \quad (9)$$

and $G_0(n,m)$ denote the number of ways that at least one slot is empty, given by

$$G_0(n,m) = \sum_{k=1}^{n} (-1)^{k+1} \prod_{j=0}^{k-1} (n-j)(n-k)^m \frac{1}{k!} \quad (10)$$

2.3 Operation Algorithm

The operation algorithm employs dynamic slot allocation and configures the length of selection bits depending on estimated contention level. The operation algorithm is as follows:

1. Start with initial slot size S_0 and selection bit size s_0 bits
2. Dynamically adjust the slot size using (5).
3. Decrease the selection bit size by one if O_n is less than a threshold $\overline{S}_t$. But it never goes below one.
4. Repeat the above steps until the selection bit size reaches one and O_n is estimated zero.

3 Numerical Results and Simulations

In this section, we present numerical results of the proposed probabilistic models for the analysis of *identification time*. In addition simulation results, based on the

simulation model capturing all the detailed interactions between the reader and the tags, are presented for performance analysis of EPC anti-collision protocol standard. We consider a single reader and several tags under ideal channel conditions (i.e., error-free). Table 1 summarizes the values of the various parameters in simulation.

Table 1. Parameter values used in the simulations

Parameters	Symbol	Value
Link Frequency	LF	320 KHz
Frequency tolerance	FT	±19 %
Divide ratio	DR	8
Number of subcarrier cycles per symbol	M	1
R → T Data Rate		80 Kbps
T → R Data Rate		320 Kbps
Tari		12.5 μs
Link pulse-repetition interval	T_{pri}	3.125 μs
The average length of a R→T data symbol	Pivot	18.75 μs
Reader to Tag calibration symbol	RTcal	37.5 μs
Tag to Reader calibration symbol	TRcal	25 μs
Preamble		87.5 μs
Frame-Sync		62.5 μs
Select command time		887.5 μs
Query command time		500 μs
ACK command time		400 μs
QueryRep command time		137.5 μs
RN 16 bit time (T → R)		137.5 μs
PC + EPC + CRC16 (T → R)		387.5 μs
Time from Reader transmission to Tag response	T_1	37.5 μs
Time from Tag response to Reader transmission	T_2	62.5 μs
Time a Reader waits, after T_1, before it issues another command	T_3	40 μs
Minimum time between Reader commands	T_4	75 μs

Figure 2 shows how the *identification time* varies for increasing number of tags. These results are based on the analytic models in Section 2. The identification time is defined as an nth round that satisfies $P_n(0) \leq \alpha$, where α is chosen as 0.95 and 0.99 for illustration. For a given α, the selection bit size varies from one to two. Obviously, higher α requires more rounds, thereby leading to longer latency. Interestingly, it has been observed that selection bit size is critical in determining the identification time. At the points of smaller population (10-50 tags), one-bit selection bits bring more tags to join the contention and those tags can be read without taking a significant latency. However contention becomes more critical as population size increases (more than 50 tags). Thus one-bit selection bit has a degraded performance compared with two-bit cases. These results demonstrate that careful configuration of selection bit size is required depending on contention level, especially in static allocation.

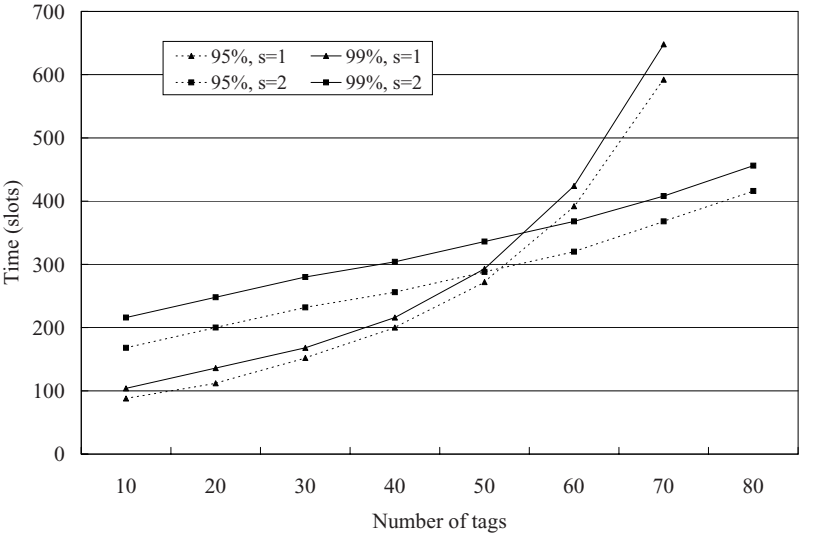


Fig. 2. Identification time versus the number of tags for different values of α and s

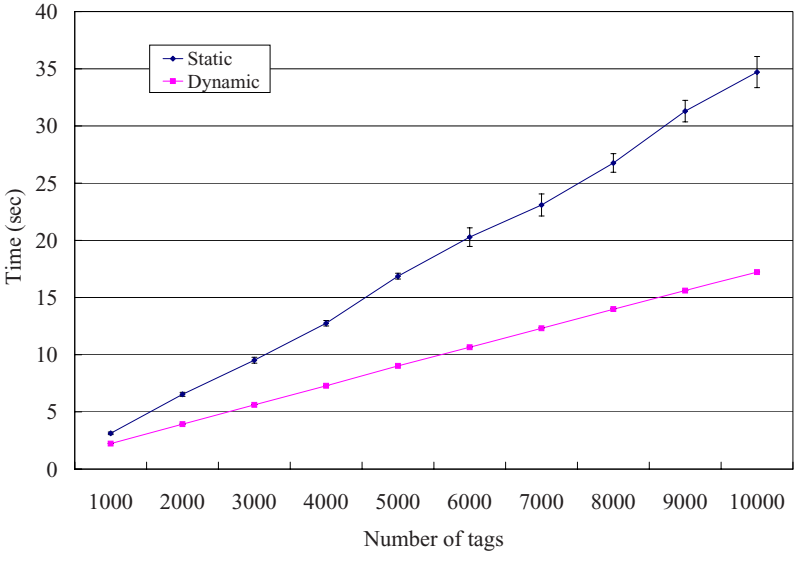


Fig. 3. Identification time versus the number of tags for static and dynamic slot allocation ($s = 5$ for the dynamic case)

Figure 3 and 4 shows the simulation results for the EPC standard using the parameters in Table 1. Identification time for increasing number of tags is shown in Fig. 3 for both operation principles. For the sake of fair comparison, static

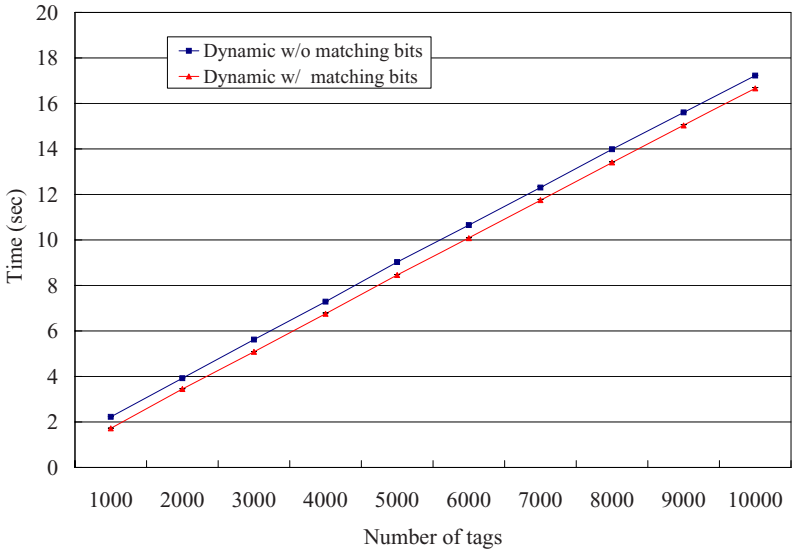


Fig. 4. Identification time versus the number of tags for fixed and dynamic matching bits ($s_0 = 5, \overline{S}_t = 32$ for the dynamic case)

allocation has the slot size that is roughly equivalent to the number of contending tag size given initial tag population.

As shown in the figure, identification time linearly grows with increasing tag population. Identification time of dynamic allocation corresponds to almost 50% of static allocation. Further investigation gives the observation that static allocation encounters depletion of tag population involved in the contention process. Figure 4 shows the impacts of adjustment of selection bit size on identification time. Unlike the expectation great improvements was not observed though continuous performance gap is observed for overall population sizes. These phenomena can be explained as follows. For static allocation, selection bit size has significant impacts on identification time since fixed slot size and selection bits size determine the contention behavior. In contrast, dynamic allocation adjusts the number of slots according to estimated number of contending tags. Although fixed selection bits size throttles the number of contending tags as tags are being identified over time, dynamic adjustment of slot size can handle this with reduced open slots in the frame. However fixed selection bits incurs more iterations that causes a slight longer latency in identification time.

4 Conclusions

In this study we analyzed the performance of EPC Class-1 Generation-2 RFID anti-collision protocol and proposed an operation algorithm. By using the collision statistics in the previous frame, our algorithm estimates the number of

contending tags and allocates accordingly an appropriate number of slots in the current frame. Probabilistic analytic models were developed to calculate the identification time and to investigate the impacts of the key parameters (i.e., s, O_n) on system performance. Simulations were used to verify the performance of the proposed operation algorithm and the adequacy of analytic models. Experimental results showed that some parameters has discrepant impacts depending on the operation principles.

Acknowledgements

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment) (IITA-2005-C1090-0502-0012).

References

1. Finkenzeller, K.: RFID handbook, 2nd edn. Wiley, Chichester (2003)
2. Law, C., Lee, K., Siu, K.: Efficient memoryless protocol for tag identification. In: Proc. International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pp. 75–84 (2000)
3. Vogt, H.: Efficient object identification with passive RFID tags. In: LNCS: International Conference on Pervasive Computing, pp. 1854–1858 (2002)
4. Choi, H., Cha, J., Kim, J.: Fast wireless anti-collision algorithm in ubiquitous ID system. In: Proc. VTC '04. vol. 6, pp. 26–29 (2004)
5. Hush, D., Wood, C.: Analysis of tree algorithms for RFID arbitration. In: Proc. ISIT '98, pp. 107–115 (1998)
6. Barton, J., Kindberg, T.: The challenges and opportunities of integrating the physical world and networked systems. HPL Technical Report 2001-18 (2001)
7. Zhou, F., Chen, C., Jin, D., Huang, C., Min, H.: Evaluating and optimizing power consumption of anti-collision protocols for applications in RFID systems. In: Proc. ISLPED '04, pp. 357–362 (2004)
8. Kim, H., Quan, C.: Performance analytsis of anti-collision algorithms for large-scale RFID systems. Telecommunications Review 2, 274–289 (2005)
9. EPCglobal: EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHHz ver 1.0.9 (2004)

Worst-Case Evaluation of Flexible Solutions in Disjunctive Scheduling Problems

Mohamed Ali Aloulou[1] and Christian Artigues[2,3]

[1] LAMSADE – Université Paris Dauphine
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France
aloulou@lamsade.dauphine.fr,
http://www.lamsade.dauphine.fr/~aloulou/
[2] LIA – Université d'Avignon
339 chemin meinajariés, Agroparc, BP 1228, 84911 Avignon Cedex 9, France
christian.artigues@univ-avignon.fr
[3] CRT – Université de Montréal
C.P. 6128, succursale Centre-ville Montréal, QC H3C 3J7, Canada
artigues@crt.umontreal.ca

Abstract. In this paper, we consider the problem of evaluating the worst case performance of flexible solutions in non-preemptive disjunctive scheduling. A flexible solution represents a set of semi-active schedules and is characterized by a partial order on each machine. A flexible solution can be used on-line to absorb the impact of some data disturbances related for example to job arrival, tool availability and machine breakdowns. Providing a flexible solution is useful in practice only if it can be assorted with an evaluation of the complete schedules that can be obtained by extension. For this purpose, we suggest to use only the best case and the worst case performance. The best case performance is an ideal performance that can be achieved only if the on-line conditions allow to implement the best schedule among the set of schedules characterized by the flexible solution. In contrast, the worst case performance indicates how poorly the flexible solution may perform. These performances can be obtained by solving corresponding minimization and maximization problems. We focus here on maximization problems when a regular min-max objective function is considered. In this case, the worse objective function value can be determined by computing the worse completion time of each operation separately. We show that this problem can be solved by finding an elementary longest path in the disjunctive graph representing the problem with additional constraints. In the special case of the flow-shop problem with release dates and additional precedence constraints, we give a polynomial algorithm that determines the worst case performance of a flexible solution.

1 Introduction

We consider a general non preemptive disjunctive problem in which a set of operations has to be scheduled on a set of machines, each operation requiring a

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1027–1036, 2007.
© Springer-Verlag Berlin Heidelberg 2007

single machine during its execution and each machine being able to process only one operation simultaneously. The operations are linked by simple precedence constraints that do not necessarily form chains. Such a model encompasses the standard flow-shop and job-shop models.

An important issue in scheduling concerns the support provided to the end-user(s) on line schedule execution after the off line scheduling phase which consists in providing an optimal or suboptimal schedule.

In disjunctive scheduling, as soon as a regular minmax objective function is considered, the support for on line scheduling often lies in providing for each machine the mandatory sequence of operations, and for each operation an earliest and a latest start time yielding operation slacks. The sequences and the time windows are such that scheduling the operations in the predetermined order and inside their time windows is feasible and keeps the objective function in a range of acceptable values. Such a flexibility provided to the end-user is referred to as temporal flexibility. This paper addresses the problem of providing more flexibility than the classical temporal one in disjunctive scheduling problems where the objective is to minimize a regular minmax objective function. As already considered in previous studies [2,3,5,6,7,12], this can be achieved by defining only a partial order of the operations on each machine, leaving to the end-user the possibility to make the remaining sequencing decisions. This is the principle of the groups of permutable operations model that has been studied by several authors [3,5,6,7,12]. The group model sets restrictions on the proposed partial orders that we relax in this paper. Indeed we represent the partial orders through general precedence constraints between operations of the same machine, that have not to be distinguished from the structural precedence constraints.

Providing a partial solution through partial orders is useful in practice only if it can be assorted with an evaluation of the complete solutions that can be obtained by extension. More precisely, given a reasonable decision policy followed by the decision maker, the following questions have to be answered. Do there remain decisions (following the decision policy) leading to a feasible schedule? What is the best and the worse objective function value reachable by the remaining set of decisions? Answering these questions provides a performance guarantee if the given on line policy is followed.

In this paper, we focus on the worst case performance evaluation. We consider disjunctive problems with a minmax objective function. We assume that the decision maker follows a semi-active schedule policy to extend the proposed partial orders. A schedule is called semi-active if the operations cannot be shifted to start earlier without changing the operation sequences or violating precedence constraints or release dates [4]. In this case, the worse objective function value can be determined by computing the worst-case completion time of each operation separately. We show that the problem of computing the worse completion time of an operation in all feasible semi-active schedules can be done by finding an elementary longest path in the disjunctive graph representing the problem with additional constraints. This gives a general framework integrating previous studies [1,3,5,6,7,12].

In the special case of the flow-shop problem with release dates and additional precedence constraints, we give a polynomial algorithm that computes the maximal completion times of all operations in all feasible semi-active schedules. These results generalize some results previously established for the single machine version of this problem [1].

In section 2, we present the problem in more details. Then, we discuss the related work in section 3. In section 4, the longest path formulation of the problem is given. In section 5, we present the polynomial algorithm for the flow-shop case. Finally, we provide a conclusion in section 6.

2 Problem Setting

We consider the following disjunctive scheduling problem. There is a set $N = \{1, \ldots, n\}$ of operations to be scheduled on m machines. m_i, p_i and r_i denote the machine, processing time and release date of operation i, respectively. The release date is the earliest time when the operation processing can start. Each operation is associated with a non-decreasing cost function $f_i(C_i)$ of its completion time C_i. We introduce two dummy operations 0 and $n+1$ such that $p_0 = p_{n+1} = 0$. This problem is represented by a disjunctive graph $\mathcal{G} = (V, C, D)$ [11]. V is the set of vertices corresponding to operations $i \in N$ and the two dummy operations 0 and $n+1$. C is the set of conjunctive arcs representing the precedence constraints between the operations. Each conjunctive arc (i, j) is valuated by p_i. D is a set of pairs of disjunctive arcs $\{(i, j), (j, i)\}$ for each pair of operations $i, j \in N$ requiring the same machine for their execution ($m_i = m_j$). We have $D = \{\{(i, j), (j, i)\} | i \neq j \text{ and } m_i = m_j\}$. Arc (i, j) represents the decision to sequence i before j, whereas arc (j, i) represents the decision to sequence j before i on the machine. In the remaining a pair of disjunctive arcs $\{(i, j), (j, i)\}$ is called a disjunction and denoted by e_{ij} or e_{ji}.

Let $\mathcal{D}$ denote the set of all disjunctive arcs, i.e. $\mathcal{D} = \{(i, j) | e_{ij} \in D\}$. A selection π is a (possibly empty) set of arcs such that $\pi \subseteq \mathcal{D}$ and $|\pi \cap e_{ij}| \leq 1$, for all $e_{ij} \in D$. Let $D(\pi) = \{e_{ij} \in D | e_{ij} \cap \pi = \emptyset\}$. A selection is complete if $D(\pi) = \emptyset$, otherwise it is partial. The disjunctive graph issued from a complete or partial selection π is denoted by $\mathcal{G}(\pi) = (V, C \cup \pi, D(\pi))$. Given a set of arcs E, let $G(E)$ denote graph $(V, C \cup E)$. A complete selection π is feasible if the graph $G(\pi) = (V, C \cup \pi)$ is acyclic. The completion time $C_i(\pi)$ of any operation $i \in N$ in the semi-active schedule derived from the complete feasible selection π is equal to the length of the longest path in $G(\pi)$ from 0 to i plus p_i. Let Π denote the set of feasible complete selections. The objective of the classical scheduling problem is to find a complete feasible selection $\pi \in \Pi$ such that a *regular* minmax objective function $F(C_1(\pi), \ldots, C_n(\pi)) = \max_{i=1,\ldots,n} f_i(C_i(\pi))$ is minimized.

Here, we assume the decision maker makes on line the remaining sequencing decisions on each machine following a semi-active policy until obtaining a complete selection π. A feasible semi-active schedule can be obtained by a list scheduling algorithm as soon as C is acyclic : sort the operations in a non

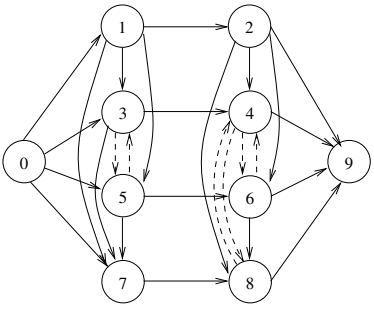


Fig. 1. The disjunctive graph for a partial selection

decreasing order of their level in $G = (V, C)$ then sequence as soon as possible on its machine each operation according to this order. The problem tackled in this paper is the following problem (SP) : Given a disjunctive graph $\mathcal{G} = (V, C, D)$, what is the worst case objective function value of the feasible semi-active schedules, i.e compute $\max_{\pi \in \Pi} F(C_1(\pi), \ldots, C_n(\pi))$?

To illustrate this problem, consider the following 2-machine and 4-job flow-shop problem. This gives 8 operations and the flow-shop context sets $m_1 = m_3 = m_5 = m_7 = 1$ and $m_2 = m_4 = m_6 = m_8 = 2$. Furthermore, we have $p_1 = 1$, $p_2 = 6$, $p_3 = 2$, $p_4 = 5$, $p_5 = 4$, $p_6 = 6$, $p_7 = 6$ and $p_8 = 1$. All release dates are equal to 0 except for $r_5 = 2$. All objective functions are the completion times of the activities. Let us consider additional precedence constraints $\{(1,3),(1,5),(1,7),(3,7),(2,4),(2,6),(6,8)\}$. We obtain the disjunctive graph $\mathcal{G}$ displayed in Figure 1.

The precedence constraints restrict the possible sequences to $(1,3,5,7)$ and $(1,5,3,7)$ on machine 1 and $(2,4,6,8)$, $(2,6,4,8)$ and $(2,6,8,4)$ on machine 2. We obtain the 6 schedules displayed in Figure 2. Note that such restriction cannot be modeled by the group of permutable operation representation used in [1,5,6,7,12]. The solution of problem SP is 20 which is the worst-case makespan value of the 6 semi-active schedules. Note that the optimal makespan of the flow-shop problem is 19.

3 Literature Review

Problem SP can be viewed as a maximization problem of an objective function that is naturally minimized. Several works have aleady been proposed in this domain. In particular, Aloulou, Kovalyov and Portmann [1] adapt the traditional three-field notation $\alpha|\beta|\gamma$ to this class of problems.

They denote this family of considered maximization problems as $\alpha(sa)|\beta|(f \rightarrow \max)$. The first field α provides the shop environment. Here $\alpha \in \{1, F, J\}$ for respectively single machine (1), flow shop (F) and job shop (J) problems. *sa* indicates that we search for the worst schedule among all semi-active schedules, which correspond to the considered online scheduling policy.

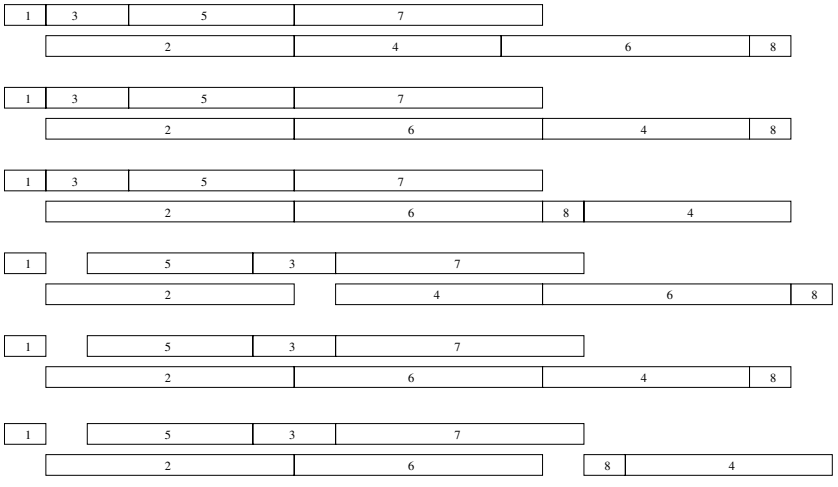


Fig. 2. Sequences and semi-active schedules compatible with the partial selection

The second field gives additional constraints on operations. The third field contains information about the criterion to maximize.

To the best of our knowledge, the first related results we are aware are due to Posner [10]. Posner studied reducibility among single machine weighted completion time scheduling problems including minimization as well as maximization problems. In these problems, the jobs may have release dates and deadlines but there are no precedence constraints between the jobs. Besides, inserting idle times between the jobs is allowed.

Aloulou *et al* [1] studied several maximization versions in a single machine environment. They examined problems $1(sa)|\beta|(\gamma \rightarrow \max)$, where $\beta \subseteq \{r_i, prec\}$ and $\gamma \in \{f_{\max}, C_{\max}, L_{\max}, T_{\max}, \sum(w_i)C_i, \sum(w_i)U_i, \sum(w_i)T_i\}$. They showed that these problems are at least as easy as their minimization counterparts, except for problems $1(sa)||(\sum w_i T_i \rightarrow \max)$ and $1(sa)|r_i|(\sum w_i T_i \rightarrow \max)$, which are still open. In particular, problems $1(sa)|r_i, prec|(L_{\max} \rightarrow \max)$ and $1(sa)|r_i, prec|(T_{\max} \rightarrow \max)$ can be solved in $O(n^3)$ times while the minimization counterparts are strongly NP-hard, even if $prec = \emptyset$ [9].

This work is closely related to the work of the present paper. Indeed, problem (SP) can be denoted, in the Aloulou *et al* notation, as $\alpha(sa)|r_i, prec|(f_{\max} \rightarrow \max)$. Besides, we propose in section 5 an algorithm solving the flow-shop problem $F(sa)|r_i, prec_k|(f_{\max} \rightarrow \max)$, generalizing the algorithm of Aloulou *et al* for problem $1(sa)|r_i, prec|(f_{\max} \rightarrow \max)$ [1]. Here $prec_k$ denotes precedence constraints appearing only between operations scheduled on the same machine, besides the classical flow-shop precedence constraints.

Another class of related work in the context of flexibility generation for online scheduling is linked to the concept of groups of permutable operations [5,6], also called ordered group assignment [3,12]. A group of permutable operations is a restriction of the sequential flexibility considered here in such a way that

each operation is assigned to a group and, there is a complete order between the groups of operations performed on the same machine. There are no precedence constraints between the operations of the same group. Heuristics have been designed to generate groups of permutable operations for general disjunctive problems [5,6] and multiobjective methods have been designed to find a compromise between flexibility and performance in the two-machine flowshop [7]. Artigues *et al* [3] propose a polynomial algorithm to perform the exact worst-case evaluation of an ordered group assignment. This method is based on longest path computations in a so-called worst-case graph, derived from the considered ordered group assignment. This method solves problem SP for general disjunctive problems (e.g. job-shop) where the disjunctions appear only between operations of the same group (inside each group the graph of disjunctions e_{ij} is a clique) and when the precedence constraints are defined between operations of different groups.

As an illustration, the selection π proposed for the flow shop example yielding the 6 feasible schedules of Figure 2 with a worst-case makespan of 20 cannot be represented by groups of permutable operations.

4 A Longest Path Formulation of the Problem

Let $\hat{C}_i$ denote the worst case completion time of operation i, i.e. $\hat{C}_i = \max_{\pi \in \Pi} C_i(\pi)$. Computing $\hat{C}_i$, for each $i \in N$ solves problem SP since the objective function is a minmax function of non decreasing functions of the completion times. Recall that $\mathcal{D}$ is the set of all disjunctive arcs. Let us now consider the following Constrained Longest Path problem associated to operation i ($CLP(i)$).

Definition 1. *Given a disjunctive graph $\mathcal{G} = (V, C, D)$ and an operation i, problem $CLP(i)$ consists in computing the longest elementary path $L^*(0, i)$ from 0 to i in $G(\mathcal{D}) = (V, C \cup \mathcal{D})$ such that $G(L^*) = (V, C \cup L^*(0, i))$ is acyclic.*

We have the following result.

Theorem 1. *The worst case completion time $\hat{C}_i$ is equal to the length of path $L^*(0, i)$ solution of problem $CLP(i)$.*

Proof. We first show that (a) $\hat{C}_i$ is the length of an elementary path l from 0 to i in $G(\mathcal{D})$ and that $G(l)$ is acyclic. Let $\pi \in \Pi$ such that we have $\hat{C}_i = C_i(\pi)$. π is the complete selection such that $\hat{C}_i$ is the length of a longest path l from 0 to i in $G(\pi)$. Since $G(\pi)$ is acyclic, l is elementary and since $l \subseteq \pi$, $G(l)$ is also acyclic. Since $\pi \subset \mathcal{D}$, l is also an elementary path in $G(\mathcal{D}) = (V, C \cup \mathcal{D})$.

Let us show that (b) any elementary path L from 0 to i in $G(\mathcal{D})$ verifying $G(L)$ is acyclic is such that there exists a feasible complete selection $\pi \in \Pi$ verifying $C \cup L \subseteq C \cup \pi$. Suppose that L includes only conjunctive arcs. Then $L \subseteq C$ and (b) is verified. Suppose now that L includes also disjunctive arcs. Since L is elementary, we have $|L \cap e_{ij}| \leq 1$ for each disjunction e_{ij}. Hence $L \setminus L \cap C$ is a partial selection. Furthermore, since $G(L) = G(V, C \cup L)$ is acyclic $C \cup L$ defines

a new acyclic precedence constraints graph and the disjunctive problem defined by $(V, C \cup L, D(L))$ is feasible. Hence $L \setminus L \cap C$ is included in a feasible compete selection.

From (a) and (b) it follows that, $\hat{C}_i$ is the length of $CLP(i)$-solution. □

Note that in the general case, problem $CLP(i)$ may be not easy to solve since it admits as a particular case the search for the longest elementary path in a graph with positive length cycles. This problem is known to be NP-hard for general graphs [8]. In Figure 3 we illustrate the problem and the necessity of the no-cycling condition in definition 1 for a job-shop with 2 machines, 3 jobs and no release dates. Operations 1,4 and 5 are assigned to the first machine and operations 2,3 and 6 are assigned to the second machine. Structural precendence constraints are $(1, 2)$, $(3, 4)$ and $(5, 6)$. $(3, 2)$ and $(1, 4)$ are additional precedence constraints. The longest elementary path from 0 to 7 is displayed in bold. Such a path is infeasible since it induces a cycle with precedence constraints $(3, 4)$.

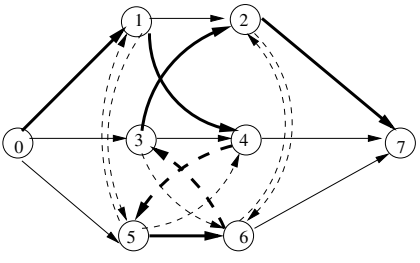


Fig. 3. A job-shop example and an infeasible elementary longest path

The following proposition shows on the opposite that the problem is simplified in a flow-shop context, since the no cycling condition is not necessary and in next Section we show that the problem is polynomially solvable.

Proposition 1. *Given a disjunctive graph $\mathcal{G} = (V, C, D)$ of a flow-shop with additional precedence constraints and an operation i, if $L(0, i)$ is an elementary path from 0 to i in $G(\mathcal{D}) = (V, C \cup \mathcal{D})$ then $G(L) = (V, C \cup L)$ is acyclic.*

Proof. Due to the flow-shop structure, the strong connected component of $G(\mathcal{D})$ include only operations assigned to the same machine. Hence any elementary cycle of $G(\mathcal{D})$ involves only operations assigned to the same machine. Let L denote an elementary path in $G(\mathcal{D})$. By definition L is acyclic and no cycle can be created by adding structural precedence constraints to L. □

5 A Polynomial Algorithm for the Flow-Shop Case

In this section, we consider the flow-shop problem with operation release dates and additional precedence constraints appearing only between operations scheduled on the same machine, as in the example presented is section 2. In this case,

any sequence of operations compatible with the precedence constraints of machine k yields a feasible complete selection. For each operation j, let j^- denote its job predecessor. We assume that if j is the first operation of its job, then j^- is a dummy operation denoted j^0. Let Γ_j^- (resp. Γ_j^+) denote the set of operations that must be scheduled before (resp. after) j on machine m_j. Let I_j denote the set of operations of machine m_j that are not linked to j with any precedence constraint. We have

$$\Gamma_j^- = \{i \neq j | m_i = m_j \text{ and there is a path from } i \text{ to } j \text{ in } G = (V, C)\} \quad (1)$$

$$\Gamma_j^+ = \{i \neq j | m_i = m_j \text{ and there is a path from } j \text{ to } i \text{ in } G = (V, C)\} \quad (2)$$

$$I_j = \{i \neq j | m_i = m_j, i \notin \Gamma_j^- \text{ and } j \notin \Gamma_i^-\} \quad (3)$$

Let us define $\hat{C}_{j^0} = r_j$. We have the following result.

Lemma 1. *The worst case completion time of any operation j is given by*

$$\hat{C}_j = p_j + \max \begin{cases} r_j, & (a) \\ \hat{C}_{j^-}, & (b) \\ \max\limits_{i \in I_j \cup \Gamma_j^-} \{\max(r_i, \hat{C}_{i^-}) + \sum\limits_{x \in I_j \cup \Gamma_j^- \setminus \Gamma_i^-} p_x\} & (c) \end{cases}$$

Proof. A recursion argument is used on machine k.

Consider machine 1, we have $\hat{C}_{j^-} = \hat{C}_{j^0} = r_j$, hence terms (a) and (b) are redundant. Denote by S the schedule in which $C_j(S) = \hat{C}_j$ and the block B of operations consecutive on machine 1, ending with j, is such that there is no idle time between any two consecutive operations in B and B is of maximal size. B always exists since we have at least $j \in B$. If $B = \{j\}$, then the starting time of j, S_j, is such that $S_j = r_j$ and (a) is verified.

If $|B| > 1$, then we have $S_j \geq r_j$. Let i be the first operation of block B. i is not a machine successor of j and $i \in I_j \cup \Gamma_j^-$. Similarly, by definition all operations inside B, except j itself, belong to $I_j \cup \Gamma_j^- \setminus \Gamma_i^-$ (they cannot be machine predecessors of i). Let us now consider the operations scheduled before i on machine 1. Let x denote the operation scheduled at the largest position before i such that $x \notin \Gamma_i^-$. This operation could be inserted right after i increasing the completion time of j which contradicts the maximality of $C_j(S)$. Hence all operations scheduled before i are in Γ_i^-. This implies that all operations of $I_i \cup \Gamma_j^- \setminus \Gamma_i^-$ are scheduled after i. Conversely, suppose that x is the operation scheduled at the smallest position after j such that x is not a successor of j, i.e. $x \notin \Gamma_j^+$. This operation could be inserted right before j, increasing the start time of j which contradicts the maximality of $C_j(S)$. Hence all operations scheduled after j are successors of j. It follows that if $S_j > r_i$ then

$$\hat{C}_j \leq \max_{i \in I_j \cup \Gamma_j^-} \{r_i + \sum_{x \in I_j \cup \Gamma_j^- \setminus \Gamma_i^-} p_x\}. \quad (4)$$

Can we have $i \in I_j \cup \Gamma_j^-$ such that $\hat{C}_j < r_i + \sum_{x \in I_j \cup \Gamma_j^- \setminus \Gamma_i^-} p_x$?
Suppose that i is such an operation. It is possible to build a feasible schedule in which all the operations before i are machine predecessors of i and the operations after j are only machine successors of j. This can be made by scheduling the operations of Γ_i^- in an order compatible with the precedence constraints within this set, then the operations of $I_i \cup \Gamma_j^+$ in an order compatible with the precedence constraints within this set, then operation j, then the operations of Γ_j^+ in an order compatible with the precedence constraints within this set. The operations on machine 2 can be scheduled in any order compatible with the precedence constraints of machine 2, and so on. In such a schedule S, we have $C_j \geq r_i + \sum_{x \in I_j \cup \Gamma_j^- \setminus \Gamma_i^-} p_x$. Hence (5.3) is verified to equality and the recusion holds for machine 1.

Suppose now the recusion holds for machine $k-1$. We can prove the recursion is then verified on machine k with simular arguments.

We first consider the schedule S in which $C_j(S) = \hat{C}_j$ and the block B of operations consecutive on machine 1, ending with j, is such that there is no idle time between any two consecutive operations in B and B is of maximal size. If $|B| = 1$ then we have either $C_j(S) = r_j + p_j$ or C_j is set by C_{j^-}. To maximize this value we have $C_j = \hat{C}_{j^-} + p_j$.

If $|B| > 1$ we can also state that an operation $i \in I_j \cup \Gamma_j^-$ starts the block with $S_i = r_i$ or $S_i = C_{i^-}$. With similar arguments as for machine 1, we prove that all operations of the set $I_j \cup \Gamma_j^- \setminus \Gamma_i^-$ are in the block. Furthermore if $S_i > r_i$ then we have $S_i = C_{i^-} = \hat{C}_{i^-}$ to have $\hat{C}_j$ maximal. Last we can show that for any operation $i \in I_j \cup \Gamma_j^-$, we can build a feasible schedule S in which all the operations before i are machine predecessors of i and the operations after j are machine successors of j and $S_i = \max(r_i, \hat{C}_{i^-})$. This achieves the recursion. □

Let $\nu = n/m$ be the number of jobs. Due to lemma 1, we have the following result.

Theorem 2. *Problem* $F(sa)|r_i, prec_k|(f_{\max} \to \max)$ *can be solved in* $O(m\nu^2)$ *times if each function* f_i *is computable in* $O(1)$ *time.*

Proof. Once sets Γ_j^- and I_j are built for each operation j, all worst-case completion times can be computed trivially via the proposed recursion by dynamic programming in $O(m\nu^2)$ where $\nu = n/m$. □

In the illustrative example of section 2, the worst case completion times are given (in the order of their computation) by $\hat{C}_1 = r_1 + p_1 = 1$ (a), $\hat{C}_3 = r_5 + p_5 + p_3 = 8$ (c), $\hat{C}_5 = r_1 + p_1 + p_3 + p_5 = 7$ (c), $\hat{C}_7 = r_5 + p_5 + p_3 + p_7 = 14$ (c), $\hat{C}_2 = \hat{C}_1 + p_2 = 7$ (b), $\hat{C}_4 = \hat{C}_7 + p_8 + p_4 = 20$ (c), $\hat{C}_6 = \hat{C}_3 + p_4 + p_6 = 19$ (c), $\hat{C}_8 = \hat{C}_3 + p_4 + p_6 + p_8 = 20$ (c).

6 Conclusion

In this paper, we proposed a longest path formulation of the problem of evaluating the worst case performance of flexible solutions in disjunctive scheduling with minmax regular objective function. A flexible solution is defined by an operation partial order on each machine. We proved that this problem is polynomial in the special case of the flow-shop problem with release dates and additional precedence constraints between operations scheduled on the same machine.

We should now focus on extension of this approach to more general problems. Unfortunately, extending this approach to the job shop is not trivial. A way to solve it is to study the complexity of the problem of finding the constrained longest elementary path in the disjunctive graph of the job-shop problem.

References

1. Aloulou, M., Kovalyov, M., Portmann, M.C.: Maximization in single machine scheduling. Annals of Operations Research 129, 21–32 (2004)
2. Aloulou, M.A., Portmann, M.-C.: An efficient proactive-reactive scheduling approach to hedge against shop floor disturbance. In: Kendall, G., Burke, E.K., Petrovic, S., Gendreau, M. (eds.) Multidisciplinary Scheduling: Theory and Applications. 1st International Conference, MISTA '03, Nottingham, UK, 13-15 August 2003. Selected Papers, pp. 13–15. Elsevier, Amsterdam (to appear, 2005)
3. Artigues, C., Billaut, J.C., Esswein, C.: Maximization of solution flexibility for robust shop scheduling. European Journal of Operational Research 165(2), 314–328 (2005)
4. Baker, K.R.: Introduction to sequencing and scheduling. Wiley, Chichester (1974)
5. Billaut, J.C., Roubellat, F.: A new method for workshop real time scheduling. International Journal of Production Research 34(6), 1555–1579 (1996)
6. Erschler, J., Roubellat, F.: An approach for real time scheduling for activities with time and resource constraints. In: Slowinski, R., Weglarz, J. (eds.) Advances in project scheduling, Elsevier, Amsterdam (1989)
7. Esswein, C., Billaut, J.C., Strusevich, V.: Two-machine shop scheduling: Compromise between flexibility and makespan value. European Journal of Operational Research 167(3), 796–809 (2005)
8. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, New York (1979)
9. Lenstra, J.K., Rinnooy Kan, A.H.G., Brucker, P.: Complexity of machine scheduling problems. Annals of Discrete Mathematics 1 (1977)
10. Posner, M.E.: Reducibility among wighted completion time scheduling problems. Annals of Operations Research, 91–101 (1990)
11. Roy, B., Sussmann, B.: Les problèmes d'ordonnancement avec contraintes disjonctives, D.S. vol. 9, SEMA, Paris, France (1964)
12. Wu, S.D., Byeon, E.S., Storer, R.H.: A graph-theoretic decomposition of the job-shop scheduling problem to achieve scheduling robustness. Operations Research 47(1), 113–124 (1999)

The Search for a Good Lattice Augmentation Sequence in Three Dimensions

Tiancheng Li and Ian Robinson

Department of Computer Science and Computer Engineering
La Trobe University, Melbourne, Australia
{t.li, i.robinson}@latrobe.edu.au

Abstract. We describe a systematic approach to the search for a good lattice augmentation sequence in three dimensions, with the index of merit of constituent lattices used as the main criterion of goodness. The result of the search can be used in the construction of an efficient automatic cubature routine for three-dimensional integrals.

1 Introduction

In the two-dimensional cubature algorithm *r2d2lri* [4] (see also [2]), Hill and Robinson demonstrate the effectiveness of using a sequence of embedded lattice rules, based on what is called a lattice augmentation sequence. *r2d2lri* uses a Fibonacci lattice as the seed lattice. Fibonacci lattices are known to be "good" lattices in the sense of high trigonometric degree of precision and high index of merit. However, Fibonacci lattices cannot readily be generalized to dimensions higher than 2. In addition, the theoretical results in [3] demonstrate that the lattices generated by repeated copying from a seed lattice having high index of merit do not necessarily have high index of merit themselves. Thus, the search for a "good" embedded sequence of lattices is not simply a search for a "good" seed lattice.

Based on the theories developed in [3], we present criteria guiding the search for a "good" augmentation sequence in three dimensions and also criteria guiding the search for "good" intermediate sequences within an augmentation sequence. The selected augmentation sequence and intermediate sequences can lead to the construction of an automatic cubature routine for three-dimensional integration.

2 Lattices and Lattice Augmentation Sequences

Definition 2.1. *A rank-1 integration lattice in s dimensions is of the form*

$$L_N = \bigcup_{\mathbf{w}\in\mathbb{Z}^s} \bigcup_{j=0}^{N-1} \left(\frac{j\mathbf{z}}{N} + \mathbf{w} \right), \tag{1}$$

where $\mathbf{z} = (z_1, z_2, ..., z_s) \in \mathbb{Z}^s$, z_1, z_2, ... , z_s *and* N *are relatively prime,* $0 < z_i < N$, $i = 1, 2, ..., s$. L_N *is called a rank-*1 *simple integration lattice if* $z_1 = 1$. *We refer to* N *as the order of the lattice.*

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1037–1045, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In this paper, we will assume that L_N is a rank-1 simple integration lattice and $N > 1$ is odd.

Definition 2.2. *$L_N^\perp$ is the dual lattice of L_N.* $\mathbf{u} \in L_N^\perp \Leftrightarrow \forall \mathbf{q} \in L_N, \mathbf{u} \cdot \mathbf{q} \in \mathbb{Z}^s$.

Definition 2.3. *A lattice rule based on L_N is the cubature rule*

$$Qf = \frac{1}{N} \sum_{j=0}^{N-1} f(\mathbf{x_j}) \cong \int_{[0,1]^s} f(\mathbf{x}) d\mathbf{x}, \tag{2}$$

where $\mathbf{x_j}$, $j = 0, 1, ..., N-1$, are the distinct points in $L_N \cap [0,1)^s$.

There are several measures of goodness of a lattice for use in a lattice rule. The following two common measures relate to the dual lattice and the distance of its points from the origin. These points indicate which coefficients from the Fourier expansion of the integrand remain in the error term when the cubature rule (2) is used to estimate the integral (see [6]).

Definition 2.4. *The index of merit (IOM) of the lattice L_N is given by*

$$\rho(L_N) = \min_{\mathbf{p} \in L_N^\perp \setminus \{0\}} \left(\prod_{i=1}^{s} \bar{p}_i \right), \text{ where } \bar{v} = \begin{cases} 1, & v = 0 \\ |v|, & v \neq 0 \end{cases}. \tag{3}$$

Definition 2.5. *The trigonometric degree of precision (TDOP) of the lattice L_N is given by*

$$TDOP(L_N) = \min_{\mathbf{p} \in L_N^\perp \setminus \{0\}} \left(\sum_{i=1}^{s} |p_i| \right) - 1.$$

Sloan and Joe [6] define an n^s-copy lattice rule, which is obtained by subdividing the unit cube into n^s smaller cubes each of side $\frac{1}{n}$, and then applying an appropriately scaled version of the lattice to each smaller cube. Following Proposition 10.2 in [6], we define an $(n^k)^s$-copy lattice of a rank-1 simple lattice as follows.

Definition 2.6

$$\begin{aligned} L_{(n^k)^s N} &= \bigcup_{k_1=0}^{n^k-1} \cdots \bigcup_{k_s=0}^{n^k-1} \bigcup_{\mathbf{x} \in L_N} \left(\mathbf{x} + \frac{(k_1, ..., k_s)}{n^k} \right) \\ &= \bigcup_{\mathbf{w} \in \mathbb{Z}^s} \bigcup_{k_1=0}^{n^k-1} \cdots \bigcup_{k_s=0}^{n^k-1} \bigcup_{j=0}^{N-1} \left(\frac{j\mathbf{z}}{N} + \frac{(k_1, ..., k_s)}{n^k} + \mathbf{w} \right). \end{aligned}$$

This definition can be interpreted as the result of shifting the lattice a distance of $\frac{1}{n^k}$ in each of $2^s - 1$ directions, then combining these shifted lattices with the original lattice.

It is important to note that there are no duplicated points resulting from the copying process if N and n are co-prime. Disney and Sloan [1] point out that a

maximal rank cubature rule obtained by n^s-copying can be as good as a rank-1 cubature rule of the same order. With this in mind, using n^s-copy cubature rules derived from rank-1 cubature rules has two advantages over constructing a rank-1 cubature rule directly. Firstly, the computation needed to search for a good (n^s)-copy lattice is much less than that needed to search for a good rank-1 lattice of the same order. Secondly, intermediate-rank lattices obtained from copying a rank-1 lattice to a maximal rank lattice provide data that can be used for error estimation. Sloan and Joe [6] argue that on the grounds of both theory and practice, the best value to choose for n is $n = 2$. Accordingly, we restrict our discussion to sequences of $(2^k)^s$-copy lattices in the remainder of this paper.

Definition 2.7. *We call the lattice $L_{(2^k)^s N}$ the k^{th}-copy lattice based on L_N and denote it by $L_N(k)$, $k = 0, 1, 2, ...$, where $L_N(0) = L_N$. Then $\{L_N(0), L_N(1), L_N(2), ..., L_N(k), ...\}$ is an augmentation sequence, L_N is the seed lattice, $L_N(k)$, $k \geq 1$ is an augmented lattice and the process from $L_N(k)$ to $L_N(k+1)$ is an augmentation cycle. We denote the IOM of $L_N(k)$ by $\rho_k = \rho(L_N(k))$, $k = 0, 1, 2, ...$ Note that $\rho_0 = \rho(L_N)$.*

From Definition 2.6, we know that the augmentation sequence $\{L_N(0), L_N(1), L_N(2), ..., L_N(k), ...\}$ is an embedded sequence, i.e. $L_N(j) \subset L_N(k) \Longleftrightarrow j < k$.

Definition 2.8. *We denote the factor by which the IOM of the k^{th} member of an augmentation sequence increases over that of the $(k-1)^{th}$ member by*

$$\sigma_k = \frac{\rho_k}{\rho_{k-1}}, \quad k = 1, 2, ... \tag{4}$$

Definition 2.9. *An intermediate lattice is a lattice L, s.t. $L_N(k) \subset L \subset L_N(k+1)$, $k \geq 0$.*

Definition 2.10. *An intermediate sequence is an embedded sequence consisting of intermediate lattices and augmented lattices.*

A possible disadvantage of using a lattice augmentation sequence in practice is the exponential growth in the number of lattice points in successive sequence members. For this reason, the use of such sequences is likely to be limited to relatively low dimensions. In addition, intermediate lattices within the augmentation cycles may be used in an automatic routine based on the sequence to offset this disadvantage.

3 Choice of Seed Lattice

Our starting point in the search for a good augmentation sequence in practice is based on the outcomes of extensive testing of the two-dimensional routine *r2d2lri*. For all but the most difficult integrals, high accuracy can be attained by that algorithm by the time the lattice order reaches $N = 5696$. Using double precision arithmetic, rounding error limits the ability to attain greater accuracy beyond this order (see [4]).

Given that the three-dimensional algorithm to be designed will use the same periodizing function as *r2d2lri* (the majority cause of the rounding error), and that it will be written for a standard double precision environment, we use this observation to conservatively estimate that, in three dimensions, a reasonably accurate result can be achieved within approximately $N = (5695)^{\frac{3}{2}} = 429,888$ integrand evaluations. Accordingly, we seek good augmentation sequences in which the final augmented lattice in the sequence has approximately M points, where $M = 430,000$.

With such a starting point decided, a good augmentation sequence in terms of IOM for use in an automatic cubature routine should have the following features:

1. There are sufficient members of the sequence prior to the order of $L_N(k)$ reaching approximately M to provide sufficient scope for practical error estimation and for the possible incorporation of extrapolation into the algorithm.
2. Individual member lattices have high IOM, especially after the first one or two augmentation cycles.
3. Sizeable increases in IOM exist between consecutive member lattices in the sequence. This aids the effectiveness and soundness of practical error estimation based on differences between successive cubatures.

An additional reason for requiring a reasonable number of member lattices in the sequence is that the search for a seed lattice with a large number of points is prohibitively expensive in terms of computer time.

The results developed in [3] indicate that the above features cannot be optimized simultaneously, and so some compromises must be made. With this in mind, we commence our systematic search for a good augmentation sequence.

Step 1. Based on experimental experience with *r2d2lri*, set the point number of the final lattice in the augmentation sequence to be approximately M. The point number of the seed lattice and the number of augmentation cycles can then be decided accordingly.

Step 2. There are four essentially different options for the choice of seed lattice.

1. If a seed lattice with $N \cong 53736$ is used, then the point number becomes approximately M after one augmentation cycle.
2. Using $N \cong 6717$ will lead to approximately M points after two augmentation cycles.
3. Approximately M points are reached after three augmentation cycles if $N \cong 840$.
4. Finally, four augmentation cycles are possible if $N \cong 105$.

Theoretically, the fewer augmentation cycles used before the lattice order reaches M, the higher the index of merit of the final augmented lattice. However, not only does a small number of augmentations seriously inhibit good error estimation, but it is also prohibitively expensive to search for a seed lattice leading to a good augmentation sequence when the seed lattice has such a large

number of points. For example, a search for a good augmentation sequence with just two member lattices based on a seed lattice of order $N = 6601$ took approximately two hundred hours on a desktop PC with a 2.4 GHz processor. The total search time across values of $N \in [6599, 6799]$ and all possible distinct $\mathbf{z}$'s would be approximately 1 billion hours on this machine.

Theorems in [3] indicate that there is a significant improvement in IOM for only the first few augmentations in a sequence. In fact, numerical experimentation discloses that for $s = 3$ and most seed lattices, the improvement in IOM between successive augmented lattices drops to a factor of only 2 beyond about three augmentation cycles. Therefore, in what follows, we focus on the search for seed lattices that will lead to good augmentation sequences having three augmentation cycles.

Step 3. In the practical search, we consider seed lattices with point number in the range $N = 781$ to $N = 898$, since these lattices will lead to the final augmented lattice having $400,000 - 460,000$ points. The criteria for choosing a seed lattice leading to a good augmentation sequence are that $\sigma_k > 4,\ k = 1, 2$ and the ratio of total IOM of all copied member lattices against the point number of the seed lattice, $\frac{(\rho_1+\rho_2+\rho_3)}{N}$ is as high as possible. Table 1 lists all seed lattices with odd order in the range $N = 781$ to $N = 898$ such that $\frac{(\rho_1+\rho_2+\rho_3)}{N} > 11.0$. The table also shows the TDOP of these lattices.

Table 1. Seed lattices that lead to good augmentation sequences

N	$\mathbf{z}$	TDOP	ρ_0	ρ_1	ρ_2	ρ_3	$(\rho_1 + \rho_2 + \rho_3)/N$
787	(1, 239, 347)	11	42	336	2368	6296	11.44
797	(1, 292, 334)	10	42	336	2448	6376	11.49
799	(1, 244, 361)	12	48	384	2400	6392	11.48
823	(1, 230, 311)	14	40	320	2432	6584	11.34
853	(1, 254, 376)	13	49	392	2432	6824	11.31
855	(1, 164, 391)	14	56	448	2336	6840	11.26
857	(1, 251, 383)	12	45	360	2592	6856	11.44
877	(1, 273, 367)	13	48	384	2656	7016	11.47
883	(1, 258, 346)	12	42	336	2560	7064	11.28
889	(1, 239, 388)	14	40	320	2560	7112	11.24

Step 4. TDOP, which is also an important measure of quality, is used as the second selection criterion when there is insignificant difference in the value of $\frac{(\rho_1+\rho_2+\rho_3)}{N}$ for different seed lattices. Thus, considering TDOP and IOM jointly, we choose the lattice with $N = 823$ and generator vector $\mathbf{z} = (1, 230, 311)$ as the seed lattice for our 'good' augmentation sequence. The corresponding lattice rule is

$$Qf = \frac{1}{823} \sum_{j=0}^{822} f\left(\left\{ \frac{j(1, 230, 311)}{823} \right\} \right). \tag{5}$$

Table 2. Order and quality indicators for the augmentation sequence based on the lattice L_{823} with generator vector $\mathbf{z} = (1, 230, 311)$

Augmented lattice	$L_{823}(0)$	$L_{823}(1)$	$L_{823}(2)$	$L_{823}(3)$
Number of points	823	6584	52672	421376
IOM	40	320	2432	6582
TDOP	14	29	59	119

The orders and quality indicators for the chosen augmentation sequence are given in Table 2.

4 Choice of Intermediate Sequences

Having chosen a good augmentation sequence, we now turn our attention to finding good intermediate sequences between consecutive augmented lattices. Since the main use of the cubatures based on these intermediate lattices is to achieve termination of the sequence as early as possible for a given requested accuracy, we again seek intermediate lattices with high IOM. However to maximize reliability of error estimates based on cubature differences, we also need good separation of the IOM of the intermediate lattices.

For $s = 3$, there are $(2^3 - 1)(2^2 - 1) = 21$ possible intermediate sequence paths between each pair of augmented lattices, $L_N(k)$ and $L_N(k + 1)$. For our chosen augmentation sequence, we know ρ_k and ρ_{k+1}, so we are looking for good separation of IOM of intermediate lattices between $L_N(k)$ and $L_N(k + 1)$.

Inspection of the IOM of all intermediate lattices of the first three augmented cycles based on the chosen seed lattice L_{823} leads to the conclusion that the best intermediate paths, with corresponding IOM are as follows (where $\overset{x}{\Longrightarrow}$ represents a shift in the x-direction):

$$\begin{aligned}
\text{First augmentation:} \quad & 40 \overset{xz}{\Longrightarrow} 60 \overset{y}{\Longrightarrow} 84 \overset{x}{\Longrightarrow} 320, \\
\text{Second augmentation:} \quad & 320 \overset{xz}{\Longrightarrow} 480 \overset{y}{\Longrightarrow} 672 \overset{x}{\Longrightarrow} 2432, \\
\text{Third augmentation:} \quad & 2432 \overset{xy}{\Longrightarrow} 2880 \overset{xz}{\Longrightarrow} 5184 \overset{x}{\Longrightarrow} 6584.
\end{aligned}$$

Another consideration is the possibility of using extrapolation to accelerate the convergence of the sequence. For extrapolation by the ϵ-algorithm [7] to be applicable, the shift pattern between consecutive augmented lattices must be constant. With this in mind, the final intermediate sequence paths chosen are:

$$\text{First augmentation:} \quad 40 \overset{xz}{\Longrightarrow} 60 \overset{yz}{\Longrightarrow} 81 \overset{x}{\Longrightarrow} 320, \tag{6}$$

$$\text{Second augmentation:} \quad 320 \overset{xz}{\Longrightarrow} 480 \overset{yz}{\Longrightarrow} 648 \overset{x}{\Longrightarrow} 2432, \tag{7}$$

$$\text{Third augmentation:} \quad 2432 \overset{xz}{\Longrightarrow} 2688 \overset{yz}{\Longrightarrow} 5784 \overset{x}{\Longrightarrow} 6584. \tag{8}$$

Table 3. The order, IOM and TDOP of member lattices in the selected embedded sequence

	N	IOM	TDOP
L_0	823	40	14
L_1	1646	60	16
L_2	3292	81	16
L_3	6584	320	29
L_4	13168	480	33
L_5	26336	648	33
L_6	52672	2432	59
L_7	105344	2688	67
L_8	210688	5784	67
L_9	421376	6584	119

The final embedded intermediate sequence chosen is denoted as

$$\{L_0, L_1, L_2, ..., L_9\}, \tag{9}$$

where L_0 is the 823-point lattice with generator vector $(1, 230, 311)$, $L_3 = L_{823}(1)$, $L_6 = L_{823}(2)$ and $L_9 = L_{823}(3)$, and the remaining lattices are those intermediate lattices represented in (6), (7) and (8). The order, IOM and TDOP of the member lattices are summarized in Table 3.

5 Numerical Examples

The relative errors listed in Tables 3 and 4 demonstrate the effectiveness of the chosen sequence for the Gaussian integrals

$$\int_0^1 \int_0^1 \int_0^1 e^{-4.738^2(x-0.124)^2-4.814^2(y-0.207)^2-0.447^2(z-0.803)^2} dzdydx \tag{10}$$

Table 4. Relative error using the sequence listed in Table 3 to compute integral (10)

	Relative Error
L_0	3.2E-04
L_1	1.0E-04
L_2	5.5E-05
L_3	1.1E-09
L_4	3.2E-12
L_5	6.2E-14
L_6	7.1E-16
L_7	1.4E-16
L_8	1.4E-16
L_9	1.4E-16

Table 5. Relative error using the sequence listed in Table 3 to compute integral (11)

	Relative Error
L_0	2.3E-01
L_1	1.8E-01
L_2	5.6E-03
L_3	3.5E-05
L_4	5.3E-04
L_5	1.4E-06
L_6	1.1E-09
L_7	1.4E-11
L_8	4.9E-14
L_9	3.8E-16

and

$$\int_0^1 \int_0^1 \int_0^1 e^{-15.794^2(x-0.124)^2-16.048^2(y-0.207)^2-1.491^2(z-0.803)^2} dzdydx, \qquad (11)$$

where (10) is much easier than (11).

Prior to application of the lattice rules, the integrands were periodized using the sixth-order sine transformation due to Sidi [5]. This transformation has the effect of clustering many cubature points very close to the region boundary. It can also have a significant weakening effect on boundary singularities. Higher-order Sidi transformations may have even more pronounced effects, but they are more susceptible to significant rounding error problems.

The improvement in accuracy at the end of each full augmentation cycle is apparent in each case. Clearly, approximations based on the intermediate lattices could be used in a practical error estimate. Also, especially after the first augmentation, the intermediate lattices could be used to affect early termination of an augmentation sequence for a given accuracy request.

As an example of the possible use of extrapolation in a practical routine based on this sequence, applying the first step of the ϵ-algorithm to the estimates $\{L_1, L_4, L_7\}$ for integral (11) results in a new estimate with relative error 1.7E-12.

6 Concluding Remarks

We have developed criteria for a systematic search for good lattice augmentation sequences, based primarily on the index of merit of sequence members. The embedded sequence found using this approach and characterized in Table 3 forms an excellent basis for the development of an efficient automatic cubature routine for three-dimensional integrals.

Although the computer search developed in this paper is limited to dimension 3, the same methodology is also applicable and has practical value for dimension 4 or 5.

References

1. Disney, S.A.R., Sloan, I.H.: Lattice integration rules of maximal rank formed by copying rank-1 rules. SIAM Journal on Numerical Analysis 29, 566–577 (1992)
2. Hill, M., Robinson, I.: A non-adaptive algorithm for two-dimensional cubature. Journal of Computational and Applied Mathematics 112, 121–145 (1999)
3. Li, T., Robinson, I., Hill, M.: The index of merit of kth-copy lattice rules. Journal of Computational and Applied Mathematics (to appear, 2007)
4. Robinson, I., Hill, M.: An algorithm for automatic two-dimensional cubature. ACM Transactions on Mathematical Software 28(1), 73–89 (2002)
5. Sidi, A.: A new variable transformation for numerical integration. International Series of Numerical Mathematics 112, 359–373 (1993)
6. Sloan, I.H., Joe, S.: Lattice Methods for Multiple Integration. Oxford University Press, Oxford (1994)
7. Wynn, P.: On the convergence and stability of the epsilon algorithm. SIAM Journal on Numerical Analysis 3, 91–122 (1966)

Tracing Illegal Users of Video: Reconsideration of Tree-Specific and Endbuyer-Specific Methods

Hyunho Kang[1], Brian Kurkoski[2], Kazuhiko Yamaguchi[2], and Kingo Kobayashi[2]

[1] Graduate School of Information Systems
[2] Dept. of Inf. and Communications Eng.,
University of Electro-Communications, 1-5-1 Chofugaoka, Chofu-shi,
Tokyo 182-8585, Japan
{kang,kurkoski,yama,kingo}@ice.uec.ac.jp

Abstract. In our recent study, we have presented an approach for tracing illegal users in content distribution networks using watermarking and fingerprinting techniques [1][2]. In this paper we generalize our previous work, further the collusion robustness is supplemented by additional security and practical experiment. This includes a more efficient tree decision method, generalization of the fingerprinting system and detailed investigation of the robustness against collusion attacks. Content is distributed along a specified tree, with the seller as the root, and the buyers as the internal nodes or leaves. The fingerprinting step is achieved by the insertion of unique information in the video wavelet coefficients by temporal wavelet transform. Our system is able to detect the fingerprint even if the video content has been distorted by collusion attacks.

1 Introduction

In the last two decades, several protection systems have been proposed and implemented in commonly used digital distribution networks [3]. These include:

- conditional access systems for satellite, cable, and terrestrial distribution,
- digital rights management (DRM) systems for internet distribution,
- copy protection systems for distribution within digital home networks.

In this paper, we shall concentrate on tracing illegal users in DRM systems for digital video distribution. Typical uses of watermarks include copyright protection and disabling unauthorized access to content. Especially, copyright protection watermarks embed some information in the data to identify the copyright holder or content provider, while receiver-identifying watermarking, commonly referred to as fingerprinting, embeds information to identify the receiver of that copy of the content. Thus, if an unauthorized copy of the content is recovered, extracting the fingerprint will show who the initial receiver was [4]. Namely, fingerprinting is a method of embedding a unique, inconspicuous serial number (fingerprint) into every copy of digital data that would be legally sold. The buyer of a legal copy is discouraged from distributing illegal copies, which can be traced back to the last legitimate owner via the fingerprint. Fingerprinting is a passive form of security, meaning that it is effective

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1046–1055, 2007.
© Springer-Verlag Berlin Heidelberg 2007

after an attack has been applied, as opposed to active forms of security, such as encryption, which are effective from the point it is applied to when decryption takes place [3]. Although a large number of studies have been made from a cryptographic point of view [5-7], little is known about practical applications. The purpose of this paper is to address the problem of implementation of video fingerprinting.

This paper is outlined as follows. In Section 2 we describe the fingerprinting process and other considerations. Section 3 presents analysis and simulation results. Finally, Section 4 gives the conclusion.

2 Proposed Method

2.1 The Revised Summary of Our Previous Works and Point to Notice

Content is distributed along a specified tree, with the seller as the root of the tree and the legitimate users as the leaves. The internal nodes represent content buyers or sellers. Each video content has an area called the "buyer area" to embed the fingerprinting information. Because there are a limited number of buyer areas available in each tree, we propose to build sub-trees, where each sub-tree has a distinctive logo (which we call "*tree_specific step1*", see Fig. 3). In our previous paper [1][2], we will use logos which are bit-mapped images of the tree number. The extracted logo shows better performance visually using ECC (error correcting codes). ECC is integrated into our watermarking system proposed in [8].

In our experiment, a 3-level temporal wavelet transform was performed on 112 frames of video, resulting in 8 types of frames (LLL, LLH, LHL, LHH, HLL, HLH, HHL, HHH where L and H stand for low and high frequency respectively) (see Fig.1). In the experiment, R sequential frames (14 sequential frames, that is, buyer areas) from the LLH (low-low-high) frames were selected because they were found to have the minimum errors (see Fig.2, marker 'O'). The channel equivalent to the fingerprinting system was found to be a random error channel therefore we have reliable decoding using ordinary error correcting codes. This fact will give us better visibility for the extracted logo. In Fig. 1, we show each buyer's area and F_β, the frames that are used in extracting of fingerprinting information. Table 1 shows the generalized decision method of embedding areas in video content.

Table 1. Symbols used in our system, with examples

	# of sub-tree	Depth of sub-tree	Order of sub-tree	# of video frame	# of max user	Wavelet level	Buyer area	# of total user
symbol	M	d	r	f	$N=r^d$	$l=\left\lfloor \log_2 \frac{f}{N} \right\rfloor$	$(L)^{l-1}H$	$M\times N$
example	1,000	3	2	112	8	3	*LLH*	8,000
	100,000	5	4	180,000	1,024	7	*LLLLLH*	1.024 10^8

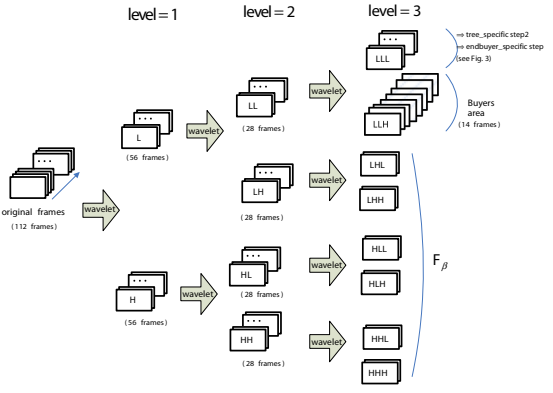


Fig. 1. Buyer areas and F_β areas after temporal wavelet transform

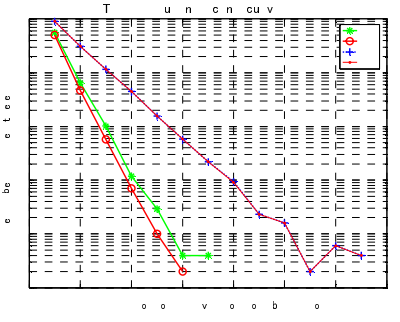


Fig. 2. Distribution of the consecutive error (y-axis in log scale)

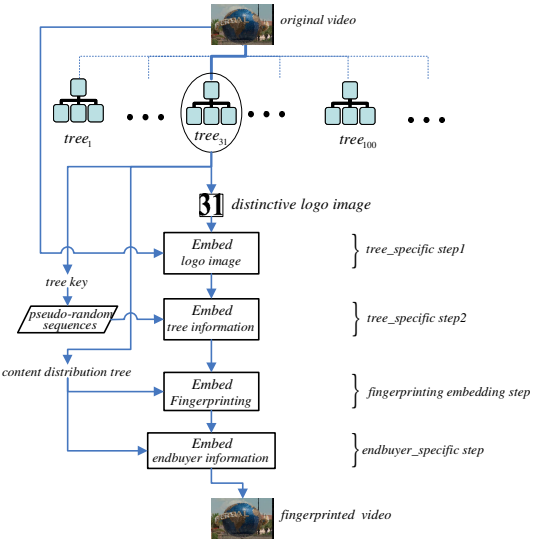


Fig. 3. The overall diagram of the embedding part

The embedding part falls into four steps which are *tree specific step1*, *tree specific step2*, *fingerprinting embedding step* and *endbuyer specific step* (see Fig. 3). In this paper, let us devote a little more attention to examining "*tree_specific step2"* and "*endbuyer_specific step*" of the overall embedding system (see Fig. 3).

A powerful attack against digital fingerprinting is the collusion attack. In our experiment, we have evaluated four kinds of collusion attacks. Those are based of the Ref. [11] which are *averaging collusion attack*, *maximum minimum collusion attack*, *negative correlation collusion attack* and *zero correlation collusion attack.*

We have written that the sub-tree scheme (which we call "*tree_specific step1*") will expand so many buyer or user in our previous papers. Within the narrow limits of the collusion attacks in only one sub-tree, it is true. But if the collusion attacks many sub-trees, it is difficult to identify the distinctive tree logo image. In this paper, we have supplemented by the embedding of tree information in first part of the fingerprinting process (which we call "*tree_specific step2*", see Fig. 3).

Especially, the most powerful attack ("*zero correlation collusion attack*") selects a fingerprinted video from a number of available fingerprinted videos. In this attack, some fingerprinting information is destroyed. However, we can trace the illegal users in our previous work as a small example. In this paper, we consider a broader range of zero correlation collusion attack. In this paper, we have supplemented by the embedding of endbuyer information in later part of the fingerprinting process (which we call "*endbuyer_specific step*", see Fig. 3).

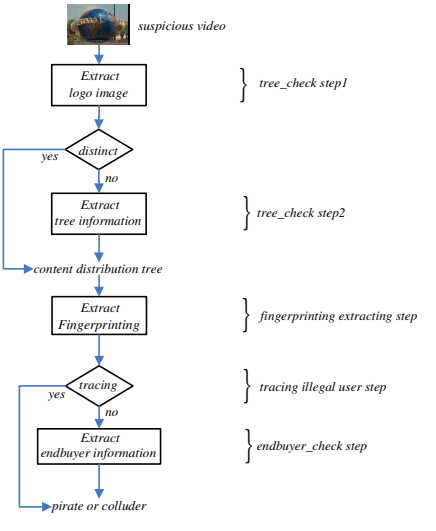


Fig. 4. The overall diagram of the detecting part

The detection falls into five steps. In this paper, we shall confine our attention to "*tree_check step2*", "*tracing illegal user step*" and "*endbuyer_check step*" in the overall detecting part (see Fig. 4).

Definition 1. Let γ (∈ Z+) be the unique ID for seller S, let k be the random number obtained from the seed ID γ, where k is a vector of floating point numbers from -1 to 1 of dimension $h \times v$ (the video frame size).

Definition 2. Let δ (∈ Z+) be the unique ID for buyer B, let p be the random permutation vector with the seed δ of dimension $h \times v$ (the video frame size).

In Fig. 5, the buyer (B) transmits the number p to the seller (S). The seller then inserts fingerprinting information $I = p(k)$ into the appropriate buyer area of the wavelet transform. When video content is distributed, fingerprinting information, I is inserted to each buyer's area of video content as described by the $tree_{31}$. Each path has a unique fingerprint. There exists a unique path between the root and user, and the unique fingerprint can be extracted to distinguish between the paths.

S_i selling to B_j inserts I_j and $\{I_a, I_b, ...\}$ into area j, where I_a, I_b, ... is the fingerprinting information for the parents of B_j in the tree. Because the video is passed hierarchically through the tree, fingerprinting information in areas a, b, ... is already present.

For example, when node-S_0 and node-B_1 engage in a transaction, fingerprinting information (I_1)—generated by the buyer and seller exchanging keys—is inserted into $buyer_1$ area of the transmitted video. When node-S_1 and node-B_4 engage in a transaction, fingerprinting information (I_1 and I_4) are inserted into $buyer_4$ area of the transmitted video. When node-S_4 and node-B_9 engage in a transaction, fingerprinting information (I_1, I_4 and I_9) are inserted into $buyer_9$ area of the transmitted video.

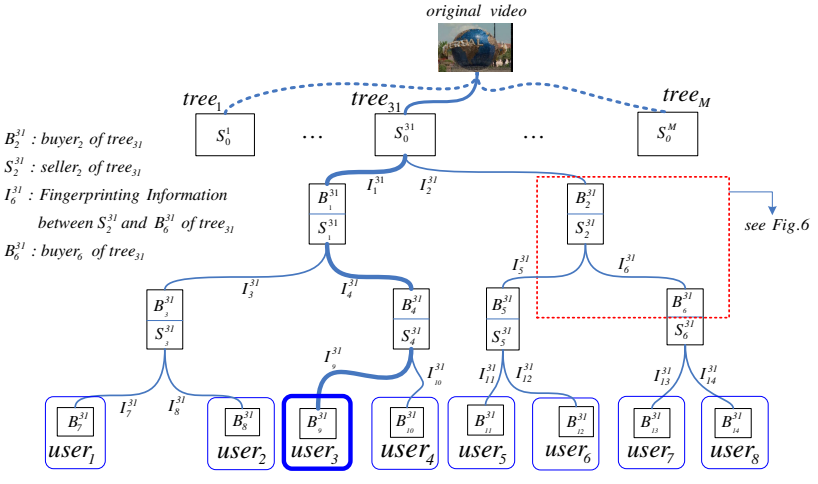


Fig. 5. Content distribution tree. Note that the number of tree was omitted in the text. If we have M sub-trees (with M logos) and N users per sub-tree, then we can support $M \times N$ users.

Therefore, whenever a seller distributes content to a buyer, different fingerprinting information is inserted. The fingerprinting information in $user_3$'s video is presented in Table 2. To detect the existence or nonexistence of fingerprinting information in illegal distributions, 196 correlation computations (14 buyers×14 areas) are required, in $tree_{31}$ for example.

Table 2. Fingerprinting information of $user_3$ video, corresponding to example in Fig. 5

buyer area	1	2	3	4	5	6	7	8	9	10	11	12	13	14
fingerprint	I_1			I_1, I_4					I_1, I_4, I_9					

Table 3. Description of the user (end buyer) – see Fig. 5

user	fingerprinting information in each buyer area
$user_1$	$I_1B_1+(I_1+I_3)B_3+(I_1+I_3+I_7)B_7$
$user_2$	$I_1B_1+(I_1+I_3)B_3+(I_1+I_3+I_8)B_8$
***user*$_3$**	$\boldsymbol{I_1B_1+(I_1+I_4)B_4+(I_1+I_4+I_9)B_9}$
$user_4$	$I_1B_1+(I_1+I_4)B_4+(I_1+I_4+I_{10})B_{10}$
$user_5$	$I_2B_2+(I_2+I_5)B_5+(I_2+I_5+I_{11})B_{11}$
$user_6$	$I_2B_2+(I_2+I_5)B_5+(I_2+I_5+I_{12})B_{12}$
$user_7$	$I_2B_2+(I_2+I_6)B_6+(I_2+I_6+I_{13})B_{13}$
$user_8$	$I_2B_2+(I_2+I_6)B_6+(I_2+I_6+I_{14})B_{14}$

Fig. 6 shows the embedding process using temporal wavelet transform, selection of the buyer's area and insertion of fingerprinting information (which we call "*fingerprinting embedding step*").

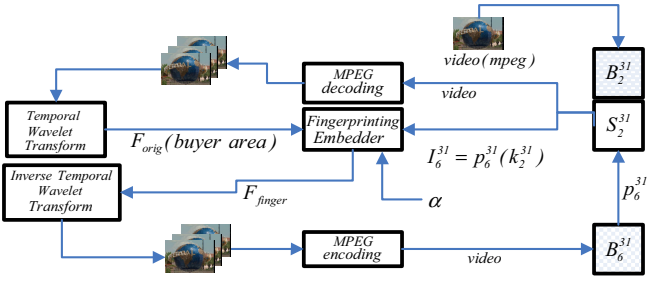


Fig. 6. Fingerprinting embedding diagram (when node-S_2 and node-B_6 engage in a transaction)

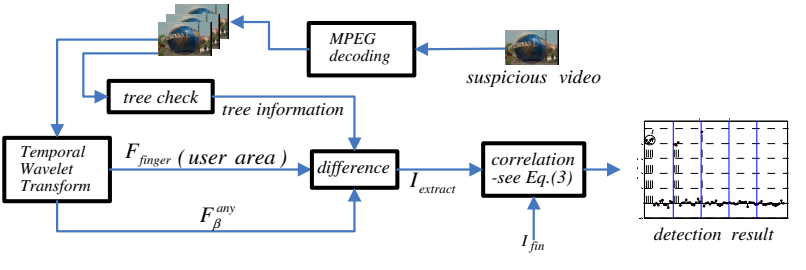


Fig. 7. Fingerprinting extracting diagram

When the content is distributed from the seller node-*S* to the buyer node-*B*, the fingerprinted video is computed using Eq. (1). The parameter α is the insertion strength; in these experiments, we choose α=0.5.

$$F_{finger} = F_{orig} + \alpha \cdot F_{orig} \cdot I_j \tag{1}$$

F_{finger} : Fingerprinted video
F_{orig} : Original video, $(L)^{l-1}H$ frames which are the buyer areas (l: wavelet level)
I_j : Fingerprinting Information (j: buyer's index).

In the extraction step, we obtain the embedded information $I_{extract}$ using Eq. (2). Note that the original video frames are not needed for the extraction step (this is blind detection).

$$I_{extract} = F_{finger} - F_{\beta}^{any} \tag{2}$$

$I_{extract}$: Extracted fingerprinting information, an estimate of the fingerprint
F_{finger} : Fingerprinted frames
F_{β}^{any} : any one frame among F_{β}
F_{β} : frames except $F_{finger}[(L)^{l-1}L]$, $F_{finger}[(L)^{l-1}H[buyer_1, ..., buyer_R]]$ (l: wavelet level).

Linear correlation is calculated by Eq. (3). Linear correlation is known to be an optimal method of detecting signals in the presence of additive, white Gaussian noise [9]. In our experiments, collusion attacks and MPEG compression appear to have AWGN characteristics. Therefore, linear correlation is suitable.

$$Cor = \frac{1}{N} \sum I_{fin} \cdot I_{extract} \quad , \tag{3}$$

where N is the video frame size ($h \times v$) and I_{fin} is $buyer_j$'s fingerprint I_j.

2.2 Considered Points

In the fingerprinting process, we have included the embedding tree information (which we call "*tree_specific step2*") and the embedding endbuyer information (which we call "endbuyer_specific step") using our previous fingerprinting method (see Eq. (1)). We use the $(L)^l$ frames as the embedding area, but the important point to notice is the controlling scaling factor α (such as less than 0.5).

In the tracing illegal user step, suppose $I_a+I_b+I_c$ exist in buyer area c. Then, I_a+I_b must exist in area b, and I_a must exist in area a. Using this rule, it is possible to reconstruct the fingerprinting in the buyer areas which were deleted by the collusion attack. We will comment on the simulation results later on.

3 Simulation Result

We have used the video sequence "universal-studio" with a frame size of 240×360 pixels and a total of 112 frames.

3.1 Fingerprinting Information Detection

To analyze the detection result, consider the content distribution tree in Fig. 5. As Fig. 8 (left) indicates, we see that fingerprint I_1 was detected in $buyer_1$ area,

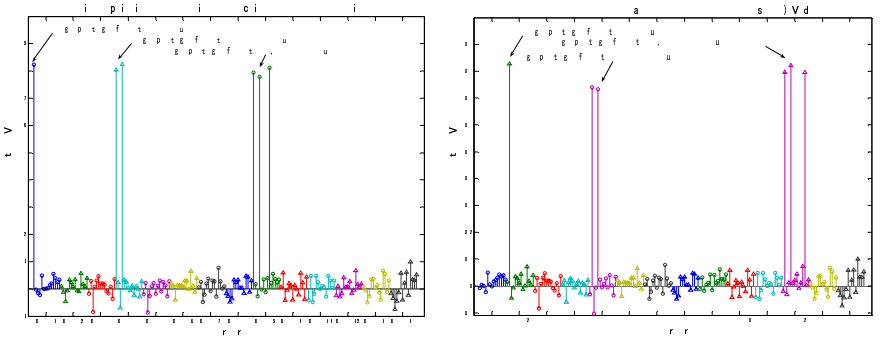


Fig. 8. Detection result from $user_3$ video (left) and $user_6$ video (right)

fingerprints I_1 and I_4 were detected in $buyer_4$ area and fingerprints I_1, I_4 and I_9 were detected in $buyer_9$ area, corresponding to the path $1 \rightarrow 4 \rightarrow 9$ for $user_3$. Thus, we can conclude that this video was distributed to end $user_3$.

As Fig. 8 (right) indicates, we see that fingerprint I_2 was detected in $buyer_2$ area, fingerprints I_2 and I_5 were detected in $buyer_5$ area and fingerprints I_2, I_5 and I_{12} were detected in $buyer_{12}$ area, corresponding to the path $2 \rightarrow 5 \rightarrow 12$ for $user_6$. Thus, we can conclude that this video was distributed to end $user_6$.

3.2 Collusion Attacks

A powerful attack against digital fingerprinting is the collusion attack. The results of our experiment show that the algorithm has some built-in resilience to collusion attacks, since the algorithm uses a long, uniformly distributed random number as fingerprinting information. In this attack, the following results were obtained.

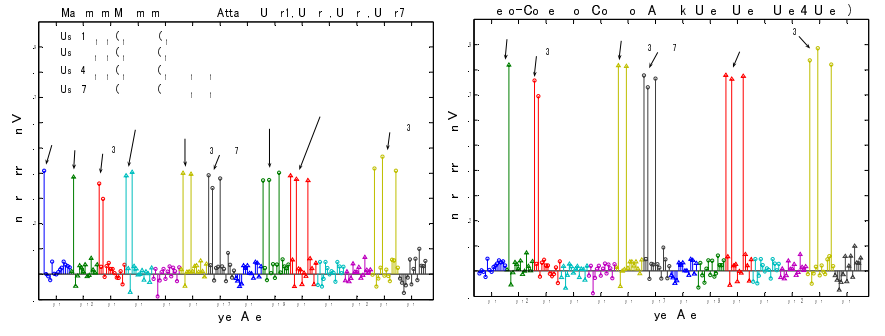


Fig. 9. Detection result after maximum-minimum (left) and zero-correlation collusion (right)

A powerful collusion attack is the maximum-minimum collusion attack proposed by Stone [10]. The attacked video is created by taking the average of the maximum and minimum values across the components of the fingerprinted video. Fig. 9 (left) correctly shows that $user_1$, $user_3$, $user_4$ and $user_7$ were colluding. The new zero

correlation attack [11] is a modification method from Stone's collusion attack. This attack selects a fingerprinted video from a number of available fingerprinted videos ($user_3$ selected as an example). In this attack, some fingerprinting information is destroyed, as shown in Fig. 9 (right). However, we know that $user_1$, $user_3$, $user_4$ and $user_7$ were colluding (see Fig. 10).

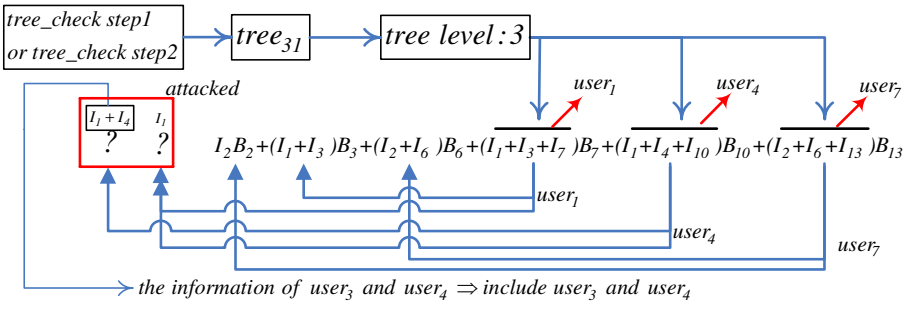


Fig. 10. Description of system considered in Fig. 9-right

We also considered the averaging collusion attack of Cox, et al. [12] and the negative correlation collusion attack which drives the correlation coefficients to a negative value [10], and obtained similar good results.

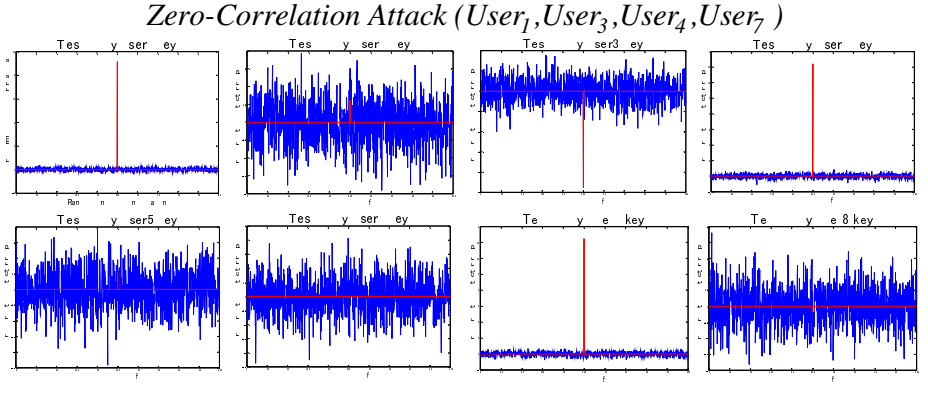


Fig. 11. Endbuyer detector response to 1000 randomly generated information. Only colluding user information matches that present in colluded fingerprinted video. (Since the result of user3 is negative correlation, we can see that the target is user3. A negative correlation is evidence of a general tendency that large values of one signal are associated with small vales of another and small values of one signal are associated with large value of another).

But, this tracing method does not necessarily apply to all cases as there are some cases where this rule does not apply. That's why we have supplemented by the embedding of endbuyer information in later part of the fingerprinting process (which we call "*endbuyer_specific step*", see Fig. 3). Due to space constraints, we do not include

the simulation results of "*tree_specific step2*" here, but we have obtained similar results of "*endbuyer_specific step*" as shown in Fig. 11.

Fig. 11 shows the response of the detector to 1000 randomly generated endbuyer information, in which the x-axis shows the 1000 independent experiments and the y-axis shows the correlation. As Fig. 11 indicates, we see that endbuyer information was detected in the colluded fingerprinted video when using the $user_1$, $user_3$, $user_4$ and $user_7$ key.

4 Conclusion

We have presented an approach for tracing illegal users in content distribution networks using video fingerprinting. Particularly, we discussed the tree specific part and endbuyer specific part for a more robust fingerprinting system. We should notice that the quality of video as controlling some scaling factors, but it may be no problem because having so many redundancies in video. The video embedding method is robust to various attacks because of the use of the temporal wavelet transform. Further research will include improvement applying with cryptographic algorithm technique.

References

1. Kang, H.H., Kurkoski, B., Park, Y.R., Lee, H.J., Shin, S.U., Yamaguchi, K., Kobayashi, K.: Video Fingerprinting System using Wavelet and Error Correcting Code. In: Song, J., Kwon, T., Yung, M. (eds.) WISA 2005. LNCS, vol. 3786, pp. 150–164. Springer, Heidelberg (2006)
2. Kang, H.H., Kurkoski, B., Yamaguchi, K., Kobayashi, K.: Tracing Illegal Users of Video Content Using Watermarking and Fingerprinting. In: SITA'05, Proceedings of the 28th Symposium on Information Theory and Its Applications, Japan, vol. 2, pp. 483–486 (2005)
3. Furht, B., Kirovski, B.: Multimedia Security Handbook. CRC Press, Boca Raton (2005)
4. Judge, P., Ammar, M.: Security Issues and Solutions in Multicast Content Distribution. IEEE Network 17, 30–36 (2003)
5. Pfitzmann, B., Schunter, M.: Asymmetric Fingerprinting. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 84–95. Springer, Heidelberg (1996)
6. Pfitzmann, B., Waidner, M.: Anonymous Fingerprinting. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 88–102. Springer, Heidelberg (1997)
7. Kundur, D., Karthik, K.: Video Fingerprinting and Encryption Principles for Digital Rights Management. Proceedings of the IEEE 92(6), 918–932 (2004)
8. Kang, H.H., Park, Y.R., Park, J.H.: Blind Watermarking based on the Spatial Domain. In: Conference on Korea Multimedia Society, Korea, vol. 5(1) (2002)
9. Cox, I.J., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan Kaufmann, Academic Press (2002)
10. Stone, H.: Analysis of Attacks on Image Watermarks with Randomized Coefficients. NEC Technical Report (1996)
11. Wahadaniah, V., Guan, Y.L., Chua, H.C.: A New Collusion Attack and Its Performance Evaluation. In: Petitcolas, F.A.P., Kim, H.-J. (eds.) IWDW 2002. LNCS, vol. 2613, pp. 64–80. Springer, Heidelberg (2003)
12. Cox, I.J., Kilian, J., Leighton, T., Shanmoon, T.: Secure Spread Spectrum Watermarking for Multimedia. IEEE Trans. On Image Processing 6(12), 1673–1687 (1997)

Rendering of Translucent Objects Based Upon PRT Techniques

Zhang Jiawan, Gao Yang, Sun Jizhou, and Jin Zhou

School of Computer Science and Technology, Tianjin University,
Tianjin 300072, P.R.China
jwzhang@tju.edu.cn

Abstract. We propose a framework for translucent objects rendering based on precomputed light transport. We apply the two-pass hierarchical technique by Jensen in the space of non-linearly approximated transport vectors, which allows us to efficiently evaluate transport vectors due to diffuse multiple scattering. Single scattering is incorporated too by considering Wang Rui's approximation. A technique from BRDF rendering is introduced to factor the phase function using a separable decomposition by Kautz to allow for decoupled precomputation and rendering. Using this approach, we achieve interactive frame rates comparable to the state-of-the-art systems.

Keywords: Subsurface scattering, phase function, separable approximation, precomputed radiance transfer, Haar wavelets.

1 Introduction

The bidirectional reflectance distribution function (BRDF) assume that light enters and exits the object surface at the same point. It's not always valid. As an example, BRDF is inadequate for translucent materials, such as marble, jade, milk, etc. To simulate the appearance of these materials, we need the more general bidirectional surface scattering reflectance distribution function (BSSRDF) (illustrated in Fig.1). To simulate the BSSRDF interactively, there are a lot of proposed methods, such as finite element methods [PB1], path tracing [HP1] [LE1], photon mapping [JH2] [JD1] and diffusion approximation [SJ1]. In the other hand, Precomputed Radiance Transfer (PRT) was first proposed by Sloan in [SP1]. In PRT, spherical harmonic (SH) function is employed for the decomposition of incident lighting and geometric scattering. It replaces the rendering equation integral by a simple set of dot products using orthogonality of spherical harmonics. rendering then reduces to the inner product of a light vector, represented in the same SH basis, with the precompiled transport vectors.

2 Related Work

In [JH3] Jensen formulated the BSSRDF as the sum of single scattering and a diffuse dipole approximation for multiple scattering. Then in [JH1] Jensen

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1056–1065, 2007.
© Springer-Verlag Berlin Heidelberg 2007

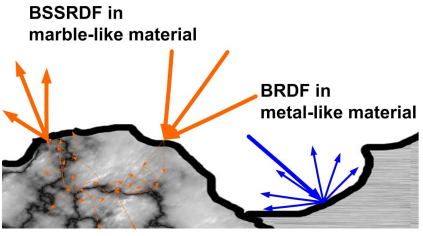


Fig. 1. Scattering of light in BSSRDF and in BRDF

presented a two-pass hierarchical integration technique to accelerate the computation of the multiple scattering component remarkably. However, previous rendering algorithms handling subsurface scattering do not allow interactive image synthesis. Several recent papers, such as [MT1] [HX1] [WR2] exploit this property and implement interactive systems to render multiple scattering. Sloan et al. included the diffuse multiple scattering component of the BSSRDF in their PRT framework[SP2], which was extended by Wang [WR1] and Liu[LX1]. One challenge of incorporating BSSRDF is to find a compact formula of light transport function that makes precomputation feasible. Sloan improved the rendering speed by compressing the matrix of spherical harmonic coefficients using clustered principal component analysis (CPCA) [SP2] .

We consider Jensen's rapid hierarchical rendering model. In multiple diffusing components, we apply their two-pass hierarchical technique in the space of transport vectors, and used wavelet approximation. In the first pass, we compute irradiance transport vectors for each selected point on the surface to sampling the irradiance. In the second pass, based upon the pre-computed irradiance samples, a rapid hierarchical evaluation of the diffusion approximation was utilized to integrate the precomputed irradiance transport vectors to compute a per-vertex multiple scattering transport vector. We use a similar approximated formula derived from [WR2] for single scattering. For an isotropic phase function, we evaluate a diffuse single scattering transport vector per vertex and combine it with the multiple scattering components. For a general phase function, we use a technique similar to [WR2]. Rendering results are showed in this paper and the appearance of translucent objects is comparable to the state of the art. This interactive system incorporates all-frequency environment lighting, reflection and subsurface scattering with both single and multiple scattering.

3 Algorithms and Implementation

BSSRDF Jensen [JH3] took the BSSRDF as the sum of a single scattering term $S^{(1)}$ and a diffuse multiple scattering term S_d:

$$S(x_i, \overrightarrow{\omega_i}; x_0, \overrightarrow{\omega_0}) = S^{(1)}(x_i, \overrightarrow{\omega_i}; x_0, \overrightarrow{\omega_0}) + S_d(x_i, \overrightarrow{\omega_i}; x_0, \overrightarrow{\omega_0}) \quad (1)$$

Approximating S_d into a dipole source model, S_d is derived as:

$$S(x_i, \overrightarrow{\omega_i}; x_0, \overrightarrow{\omega_0}) = \frac{1}{\pi} F_t(\eta_i, \overrightarrow{\omega_i}) R_d(\|x_i - x_0\|) F_t(\eta, \overrightarrow{\omega_0}) \tag{2}$$

where η is the relative index of refraction, F_t is the Fresnel transmittance and R_d is the diffuse reflectance computed by:

$$R_d(r) = \frac{a^{'}}{4\pi}[z_r(\sigma_{tr} + \frac{1}{d_r})\frac{e^{-\sigma_{tr} d_r}}{d_r^2} + z_v(\sigma_{tr} + \frac{1}{d_v})\frac{e^{-\sigma_{tr} d_v}}{d_v^2}] \tag{3}$$

where $\sigma_s^{'} = (1-g)\sigma_s$ and $\sigma_t = \sigma_a + \sigma_s^{'}$ are reduced scattering and extinction coefficients, $\sigma_a^{'} = \sigma_s^{'}/\sigma_t^{'}$ is the reduced albedo, g is the mean cosine of the scattering angle, $\sigma_{tr} = \sqrt{3\sigma_a\sigma_t^{'}}$ is the effective extinction coefficient, $d_r = \sqrt{r^2 + z_r^2}$ and $d_v = \sqrt{r^2 + z_v^2}$ are the distances from illumination point x_i to the dipole source, $r = \|x_0 - x_i\|$is the distance between x_i and x_0 , $z_r = 1/\sigma_t^{'}$ and $z_v = z_r(1+4A/3)$ are the distances from x_0 to the dipole source. Here $A = (1 + F_{dr})/(1 - F_{dr})$, and F_{dr} is a diffuse Fresnel term $F_{dr} = -1.440/\eta^2 + 0.710/\eta + 0.668 + 0.0636\eta$. This approximation is somehow expensive in implementation. Jensen [JH1] has proposed a hierarchical method which decompose compute process into two pass. In the first pass, they carried on uniformly sampling on the surface, these sample points are usually organized and stored in octree, with each node storing represented values of all its child nodes. Single scattering is computed by the following equation in which the BSSRDF single scattering term S^1 is implicitly defined:

$$L_0^{(1)}(x_0, \overrightarrow{\omega_0}) = \sigma_s \int_{2\pi} \int_0^{\infty} Fp(\overrightarrow{\omega_i}^{'}, \overrightarrow{\omega_0}^{'}) e^{-\sigma_i(s_i^{'}+s)} L(x_i, \overrightarrow{\omega_0}) ds d\overrightarrow{\omega_i} \tag{4}$$

where $\overrightarrow{\omega_i}^{'}$ and $\overrightarrow{\omega_0}^{'}$ are the refracted incoming and outgoing directions, $F = F_t(\eta, \overrightarrow{\omega_i}^{'}) \bullet F_t(\eta, \overrightarrow{\omega_0}^{'})$ is the combined Fresnel transmittance, $s_i^{'}$ and s are the scattering path lengths along $\overrightarrow{\omega_i}^{'}$ and $\overrightarrow{\omega_0}^{'}$, and p is a normalized phase function. A good approximation of $s_i^{'}$ can be found in [JH3] and [WR2]. We consider the approximation proposed by Wang [WR2]. Details are shown in Fig.2.

We use two programming language for implementation: C# and C++. The BSSRDF parts are written by C++, rendering and wavelet approximation parts are written by C# and DirectX. We assume the medium in our experiments is homogeneous and illuminated by only distant direct environment.

Multiple Scattering. The multiple scattering components in our implementation are derived from [WR2]. The illumination is assumed to be the product of incident light and visibility:

$$L(x_i, \overrightarrow{\omega_i}) = L(\overrightarrow{\omega_i}) V(x_i, \overrightarrow{\omega_i}) \tag{5}$$

where V is the visibility. The diffusing components are rearranged to:

$$L_d(x_0, \overrightarrow{\omega_0}) = \frac{1}{\pi} F_t(\eta, \overrightarrow{\omega_0}) \sum_j T_d(x_0, \overrightarrow{\omega_j}) L(\overrightarrow{\omega_j}) \tag{6}$$

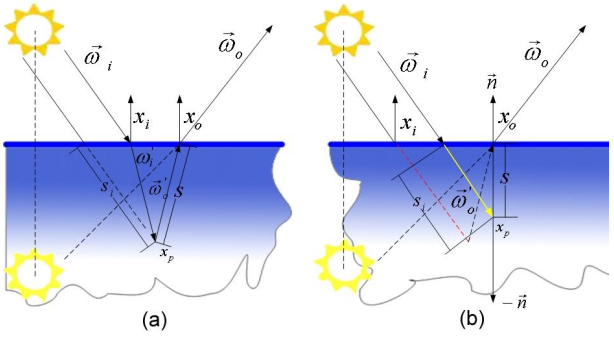


Fig. 2. Single Scattering (a) Single scattering is the result of integral along the refracted outgoing ray $\vec{\omega_0}'$. And x_p is a sample point along the integration path. The approximation of observed path length s_i. (b) The yellow solid line shows the approximated observed path length s_i, and the red dotted line is the true s_i.

$$T_d(x_0, \vec{\omega_i}) = \int_A R_d(\|x_i - x_0\|) E(x_i, \vec{\omega_i}) dA(x_i) \tag{7}$$

$$E(x_i, \vec{\omega_i}) = F_t(\eta, \vec{\omega_i}) V(x_i, \vec{\omega_i})(\vec{n_i} \bullet \vec{\omega_i}) \tag{8}$$

where T_d is the irradiance transport function to be evaluated, E is the irradiance transport function at point of illumination x_i which describes light transport from the environment to x_i , and the diffuse reflectance R_d predicts the light transport from x_i to x_0, similar to form factors in radiosity methods.

Since spherical harmonics, wavelets and L are formulated in orthonormal basis, it can apply efficiently compression, storage and rendering. Like [NR1], we use a high-resolution environment cubemap to represent an all-frequency, time-varying illumination, and approximate the environment map in a wavelet basis, keeping only the largest terms(non-linear approximation), and obtain further compression by encoding the light transport matrix sparsely but accurately in the same basis.

To compute T_d efficiently, we apply the two-pass hierarchical technique proposed by Jensen [JH1]. Specifically, in the first pass we compute irradiance transport functions E for a set of uniformly distributed surface sample points. We rasterize a high-resolution visibility hemi-cube at each x and render the raw transport matrix. We carry on a wavelet transform on these transport functions, keeping only a bounded number of the largest wavelet coefficients for approximation then dither and quantize the matrix elements to 6, 7 or 8 bits, depending on the desired level of fidelity, and discard all zero coefficients. In the second pass, we use an octree by clustering and summing the irradiance transport vectors computed in the first pass. To reduce memory consumption, we always apply wavelet non-linear compression at each node, and truncate the remaining. Like Jensen [JH1], for each vertex we traverse the octree and evaluate transport vector T_d by accumulating clustered irradiance transport vectors at each node.

Single scattering. For highly translucent materials, although it is usually multiple scattering component who dominates the appearance, single scattering also play an important role. However we face a challenge for single scattering: the integration path(the refracted outgoing direction) is unknown at precomputation time, and it is difficult to decouple the rendering from precomputation because a general phase function depends on both incident and view directions. Therefore, we considered the solution proposed by Wang et al.[WR2] which is simple but faster: we arbitrarily substitute the view-dependent integration path with a path along the opposite of normal where outgoing radiance ray lies. By doing this, we have approximated the view-dependent integration path with a view independent path known at precomputation time (see Fig.2(b)). This approximation changes the observed incident path length s_i. Then, we approximate the phase function using a separable decomposition technique introduced by [KJ1], which has been used for BRDF rendering. We can now handle phase functions in the same way as the one in [WR1] and [LX1]. The final equations for single scattering are:

$$L_0^{(1)}(x_0, \overrightarrow{\omega_0}) = F_t(\eta, \overrightarrow{\omega_0}) \sum_k h_k(\overrightarrow{\omega_0}) \sum_j T_k^{(1)}(x_0, \overrightarrow{\omega_j}) L(\overrightarrow{\omega_j}) \tag{9}$$

$$T_k^{(1)}(x_0, \overrightarrow{\omega_0}) = \int_0^\infty \sigma_s T_p(x_p, \overrightarrow{\omega_i}) e^(- \sigma_t s) ds \tag{10}$$

$$T_p(x_p, \overrightarrow{\omega_i}) = g_k(\overrightarrow{\omega_i}^{'}) F_t(\eta, \overrightarrow{\omega_i}) e^{-\sigma_t s_i^{'}} V(x_i, \overrightarrow{\omega_i}) \tag{11}$$

where $T_k^{(1)}$ is the k-th single scattering transport function, corresponding to the k-th phase function term. x_p is the sample point at distance along the integration path, and $T_p$is the illumination transport function at x describing the amount of light transported to x_p from the environment but attenuated along the incident path. Evaluation of $T_k^{(1)}$ can employ Monte Carlo techniques to integrate T_p over the sample path. For importance sampling, we pick a random distance along the sample path by $s(\xi) = -ln(1-\lambda\xi)/|\sigma_t|$, where $\xi \in [0, 1]$ is a uniformly distributed random number, $|\sigma_t|$ is the luminance of σ_t, $\lambda = 1 - e^{-|\sigma_t| d_m}$is the normalization term, and d_m is the integration upper limit, which is the maximum distance along the path. In practice, since it takes many random samples to eliminate noise, we instead use a fixed number of N deterministic sample distances by taking $s = s(\xi_i)$, where $\xi_i = (i - 0.5)/N$ and $i = 1...N$.

PRT. In general, there are two computing units on our experiment machine: CPU and GPU. In order to compute efficiently, our policy is to carry on illumination related computations on GPU and others on CPU. First, we discretize the direct light on a $8 \times 32 \times 32$ cubemap. Irradiance and illumination transport functions are rendered into the GPU. For multiple scattering, we first evaluate the irradiance transport functions for a set of evenly distributed surface sample points. At each point, we gather and store the visibility information, which will be applied to generate the transport function. We then download the data from

GPU to the CPU to perform the non-linear wavelet approximation. When irradiance sampling is completed, we apply the hierarchical integration to compute transport vectors by the diffuse approximation.

We first compute transport matrix T in the raw lighting basis of $8 \times 16 \times 16$ cubemap, then perform a singular value decomposition (SVD) on the tabulated data, and storing them as cubemap textures. Each term contains a pair of light map and view map. We apply the $K = 6$ lightmaps in precomputation and produce K transport functions per vertex. Finally, we download data to the CPU to perform the final step of non-linear wavelet approximation. GPU takes over most vector related computing work, which provides us a performance improvement.

Rendering. First, we perform a fast wavelet transform on sample of the environment map on $8 \times 32 \times 32$ cubemap. We use a 2D Haar transform here as well because Haar is an orthonormal basis. And we choose a subset of wavelet basis lights to use for the current frame for non-linear approximation. We have implemented the dot products of the light vector with the transport vectors entirely on the CPU. Since, graphics hardware is not clearly general enough to support our algorithm efficiently.

The multiple scattering components are modulated for Fresnel transmittance to determine the final vertex colors. And we incorporate K-term phase function approximation into texture lookups of the refracted view direction. Finally a simple scene was set up: the model with direct illumination stands on a glazed plane. Specular reflection was added to the model so that it looks more realistic. We use a pre-convolved environment map to add a specular component to the final rendering.

4 Results and Discussion

We conducted experiments to validate the proposed technique by using several models available at hand. First, we have run our algorithm on the Utah teapot (Fig.3). The main parameters chosen are $\sigma_s = [0.71, 0.79, 1.00]$ and $\sigma_a = [0.03, 0.38, 0.06]$. The result image is compared with the one generated by Jensen [JH1] under similar viewing and lighting conditions. The proposed approach gives similar effects with smooth speed 7.9 fps, comparable to 7.5 fps reported in [JH1]. The computation of the pre-computed integral with 300 light sources for all vertices takes about 52 minutes for the teapot model. Throughout

Table 1. Rendering results by using the proposed method

Model	Vertices	Triangles	Time	Memory	Frame Rates(fps)	
					with scattering	without scattering
Bird	31,000	28,578	20min	86MB	10.7	15.2
Teapot	150,510	292,168	52min	131MB	7.9	11.2
Smurf	372,543	310,321	121min	243MB	4.1	6.3
Dragon	437,645	871,414	135min	288MB	3.4	4.7

Fig. 3. (a) is a Utah teapot with scattering rendered by our system. (b) is an image from paper [JH1].

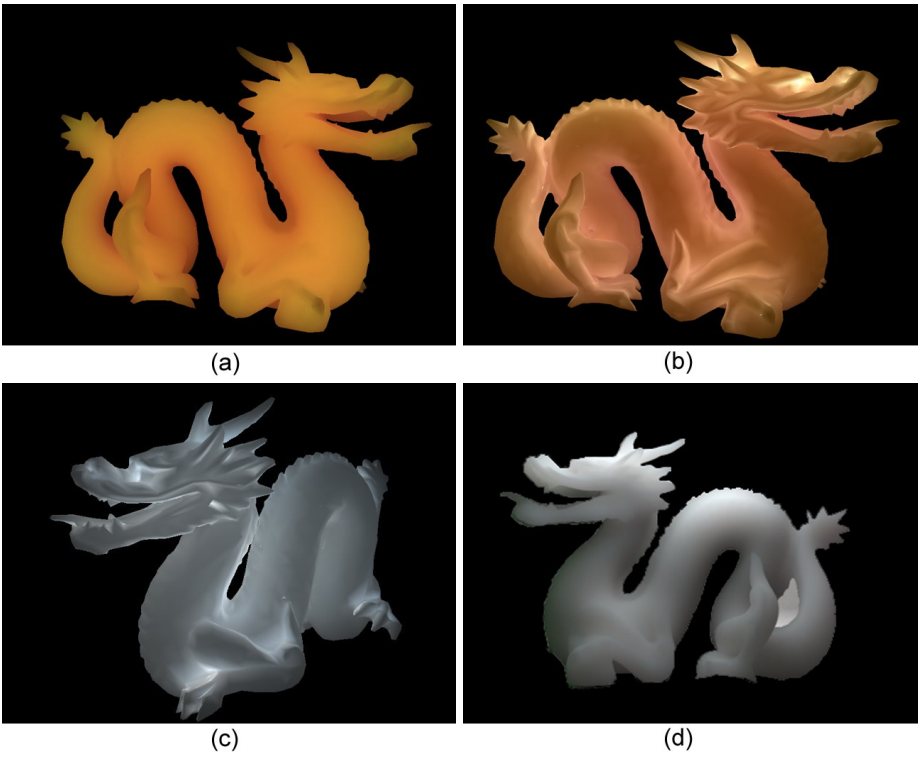


Fig. 4. (a) is a more wax-like dragon which looks soft and dim. (b) is obtained from (a) by rearrange parameters a little bit. It is much brighter than (a), and seems to be composed of plastic. Then we adjust σ_a, and obtain (c), which looks like shiny plaster. Finally, we managed to get a convincing white-jade-like dragon in (d). Notice the distinct appearance difference among them.

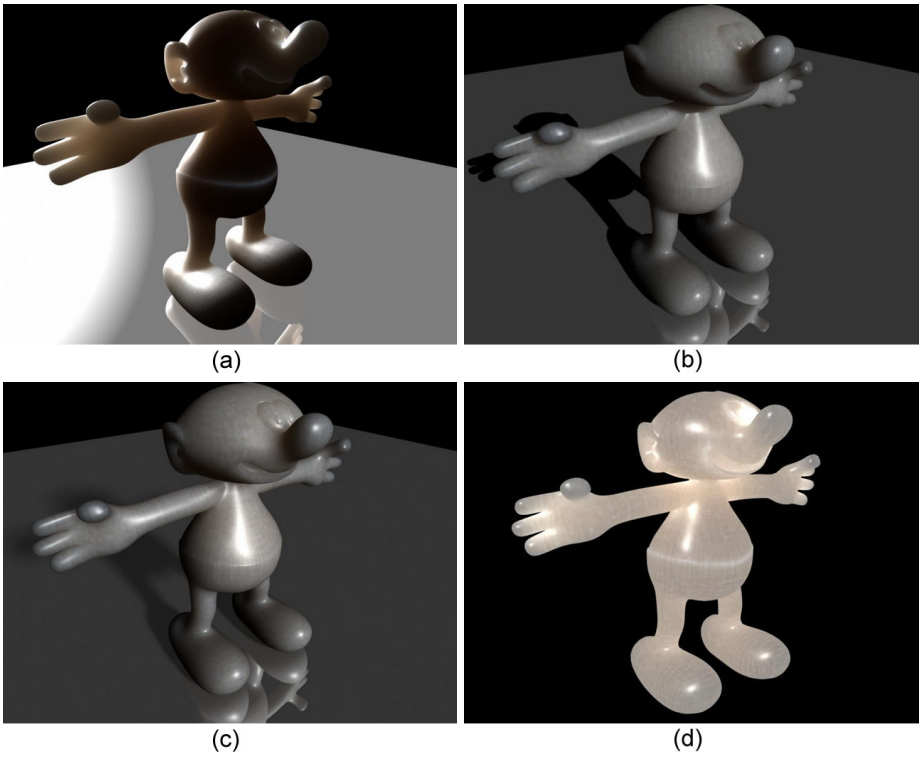


Fig. 5. A Perlin style marble texture is applied during the final rendering. In (a), smurf standing on a plane is rendered under back directional light. Note that the translucent feature around ears and figures. In addition, reflection added reinforces realistic feeling. Form (b) to (c) smurf is rendered without and with consideration of all-frequency shadow. Note that soft shadow boundaries in (c). In (d), we choose different lightmap to render smurf and obtain an interesting scene. See that it is brighter while with shadows ignored.

our experiments, we use smaller cubemap resolution of $8 \times 32 \times 32$ and keep 128(2%) wavelet coefficients. Subsurface scattering tends to blur texture and illumination, and then consequently soften the overall look of an object. Therefore compared with the BRDF, the low-frequency BSSRDF transport functions will be enough for computing. The precomputation time, memory and rendering speed are listed in Table 1. For dragon model, we tried the parameters for jade listed in some published references, but failed to gain convincing appearance of jade for our system. We experimented with a lot of parameters to find proper ones. In Fig.4, there are some kinds of dragons in our experiments. In (a), a more wax-like dragon is rendered, which looks soft and dim. In (b), deriving from (a) by rearrange parameters a little bit, it is much brighter than (a), and seems to be compose of plastic. Then we adjust σ_a , and obtain a different dragon in (c), which looks like shiny plaster. Finally, we managed to get a convincing

white-jade-like dragon in (d). Notice the distinct appearance difference among them. For smurf model, we applied a perlin style marble texture during the final rendering. In (a), smurf standing on a plane is rendered under back directional light. Note that the translucent feature around ears and fingers. In addition, reflection added reinforces realistic feeling. Form (b) to (c), smurf is rendered without and with consideration of all-frequency shadow. Note that soft shadow boundaries in (c). In (d), we choose lightmap illumination to render smurf and obtain an interesting scene. See that it is brighter while with shadows ignored.

5 Conclusion and Future Work

We have developed a system for the interactive rendering of translucent objects. The basic idea is to decompose BSSRDF from rendering, and incorporate complete Jensen's two-pass hierarchical BSSRDF model including both single and diffuse multiple scattering components into in our system. We also consider Wang's approximation for single scattering in our system. In the future we would like to show the applicability of the presented system to heterogeneous and more complicated materials. The system now is only appropriate for simple objects, not for more complicated ones. This is because in more complex objects, the distance of two surface patches on the object is not simply their Euclidean distance, and it might be much longer depending on its complexity. So some delicate techniques, such as multi-layered models, will be investigated and researched later. We also think it may be worthwhile to investigate which parts of the rendering could be computed directly on the graphics board. The implementation for dot products of the light vector with the transport vectors may be feasible to migrate to GPU for performance gains.

References

[PB1] Blasi, P., Le Saec, B., Schlick, C.: A rendering algorithm for discrete volume density objects. Computer Graphics Forum 12(3), 201–210 (1993)

[JD1] Dorsey, J., Edelman, A., Jensen, H.W., Legakis, J., Pedersen, H.K.: Modeling and Rendering of Weathered Stone. In: Proceedings of SIGGRAPH'99, pp. 225–234 (1999)

[HP1] Hanrahan, P., Krueger, W.: Reflection from layered surfaces due to subsurface scattering. In: Kajiya, J.T. (ed.) Computer Graphics (SIGGRAPH '93 Proceedings), ACM SIGGRAPH, vol. 27, pp. 165–174 (1993)

[HX1] Hao, X., Varshney, A.: Real-time rendering of translucent meshes. ACM Trans. Graph. 23(2), 120–142

[JH1] Jensen, H.W., Buhler, J.: A rapid hierarchical rendering technique for translucent materials. ACM Trans. Graph. 21(3), 576–581 (2002)

[JH2] Jensen, H.W., Christensen, P.H.: Efficient simulation of light transport in scences with participating media using photon maps. In: Proc. of SIGGRAPH '98, pp. 311–320 (1998)

[JH3] Jensen, H.W., Marchner, S.R., Levoy, M., Hanrahan, P.: A practical model for subsurface light transport. In: Proc. of SIGGRAPH '01, pp. 511–518 (2001)

[KJ1] Kautz, J., Mccool, M.D.: Interactive rendering with arbitrary brdfs using separable approximations. In: Proc. of the 10th Eurographics Rendering Workshop, pp. 281–292 (1999)

[LE1] Lafortune, E.P., Willems, Y.D.: Rendering participating media with bidirectional path tracing. In: Proc. of the 7th Eurographics Rendering Workshop, pp. 91–100 (1996)

[MT1] Mertens, T., Kautz, J., Bekaert, P., Seidelz, H.-P., Reeth, F.V.: Interactive rendering of translucent deformable objects. In: Proc. of the 14th Eurographics Symposium on Rendering, pp. 130–140 (2003)

[NR1] Ng, R., Ramamoorthi, R., Hanrahan, P.: All frequency shadows using non-linear wavelet lighting approximation. ACM Trans. Graph. 22(3), 376–381 (2003)

[SP1] Sloan, P., Kautz, J., Snyder, J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In ACM Trans. Graph. 21, 527–536 (2002)

[SP2] Sloan, P., Hall, J., Hart, J., Snyder, J.: Clustered principal components for precomputed radiance transfer. ACM Trans. Graph, 382–391 (2003)

[SJ1] Stam, J.: Multiple scattering as a diffusion process. In: Proc. of the 6th Eurographics Rendering Workshop, pp. 41–50 (1995)

[WR1] Wang, R., Tran, J., Luebke, D.: All-Frequency Relighting of Non-Diffuse Objects using Separable BRDF Approximation. In: Proc. of the 15th Eurographics Symposium on Rendering, pp. 345–354 (2004)

[WR2] Wang, R., Tran, J., Lrebke, D.: All-Frequency Interactive Relighting of Translucent Objects with Single and Multiple Scattering. In: Proc. of SIGGRAPH 2005, ACM Trans. Graphics, vol. 24(3), ACM Press, New York (2005)

[LX1] Liu, X., Sloan, P., Shum, H.-Y., Snyder, J.: All-Frequency Precomputed Radiance Transfer for Glossy Objects. In: Proc. of the 15th Eurographics Symposium on Rendering, pp. 337–344 (2004)

An Image-Adaptive Semi-fragile Watermarking for Image Authentication and Tamper Detection

Hengfu Yang[1,2], Xingming Sun[1], Bin Wang[3], and Zheng Qin[3]

[1] School of Computer and Communication, Hunan University
410082 Changsha Hunan, China
hengfuyung@etang.com, sunnudt@163.net
[2] Computer Science Department, Hunan Institute of Engineering
411104 Xiangtan Hunan, China
hengfuyung@etang.com
[3] College of Software, Dongguan University of Technology,
523106, Dongguan, Guangdong, China
qz88@263.net

Abstract. By taking full advantage of the masking characteristics of the HVS (Human Visual System), this paper first deduces a JND (Just Noticeable Differences) threshold matrix based on blocks, and then presents an image-adaptive semi-fragile watermarking technique operating in DWT (Discrete Wavelet Transform) domain. The proposed technique exploits HVS characteristics and the polarity of the sum of the absolute values of coefficients of the vertical and the horizontal detail subbands of image sub-blocks to embed a binary watermark with visually recognizable pattern. The embedded watermark is invisible to human eyes and adaptive to the original image characteristics because of the utilization of HVS characteristics. The experimental results show that the proposed algorithm is effective and can classify effectively the attacks made on the watermarked image into malicious attacks and non-malicious attacks by the adaptive threshold. The embedded watermark is sensitive to malicious attacks while is robust to non-malicious attacks. Moreover, the scheme has a good accuracy in locating tampered blocks.

1 Introduction

The concept of fragile watermark appeared as early as in 1994, however, not until 1997 did fragile watermark arouse the interests of researchers in different fields[1]. The fragile watermark will be destroyed even if the change in the marked digital media is minute. A prospective fragile watermark application is image authentication. As a tradeoff of robustness and fragility, semi-fragile watermark that can resist "content preserving" operations (such as JPEG compression) and be sensitive to "content altering" transforms (such as feature replacement) is more practicable than fragile watermark in image authentication[2]. To acquire an optimum balance among robustness, invisibility and fragility, an effective fragile watermark scheme should take into account the effect of the HVS. Some existing fragile watermarking algorithms[3-6] did not consider the effect of HVS, and others did not take full advantage of the HVS

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1066 – 1075, 2007.
© Springer-Verlag Berlin Heidelberg 2007

masking characteristics[7,8]. Reference [10] proposed an effective authentication watermarking algorithm, but the method needed a predetermined threshold. In this paper, we first gave a visual model based on blocks, and then proposed an image-adaptive semi-fragile watermarking technique based on DWT domain, which can distinguish effectively malicious attacks and non-malicious attacks by the adaptive threshold, and locate the tampered region accurately.

2 The Proposed Algorithm

2.1 Human Visual System

In order to design an effective semi-fragile watermarking, it is a necessity to take into account the visual effect of embedding a watermark into a host image. Yang et al[9] employs the HVS characteristics to design a content based image public watermarking. The assumptions are that:

(i) Human eyes have different sensitivity to different luminance, most sensitive to middle level luminance usually, Weber ratio keeps const 0.02 within a large range of middle level gray, and sensitivity declines nonlinearly within the low and high luminance range. We can use the same formula in [9], and use $\omega(u,v)$ as contrast sensitivity factor.

$$\omega(u,v)=\begin{cases}\dfrac{(\beta-0.02)\left[ave(u,v)-I_1\right]^2}{I_1^2}+0.02, & if \quad ave(u,v)\le I_1\\ 0.02, & if \quad I_1<ave(u,v)\le I_2\\ \dfrac{(\beta-0.02)\left[ave(u,v)-I_2\right]^2}{(255-I_2)^2}+0.02, & if \quad ave(u,v)>I_2\end{cases}. \tag{1}$$

whereβdenotes the maximum of contrast sensitivity, $B_{u,v}$ denotes sub-block corresponding to position (u,v) in the original image, $ave(u,v)$ is the average luminance of $B_{u,v}$, I_1 and I_2 are the predetermined gray threshold value.

(ii) Human eyes are more sensitive to the noise in image smooth areas, and less sensitive to the one in image texture areas. Let $H(u,v)$ be the entropy of sub-block $B_{u,v}$. more entropy value is corresponding to the image texture or edge area, while less entropy value corresponds to image smooth area., so we can use the entropy $H(u,v)$ to depict the texture characteristics of the sub-block $B_{u,v}$.

Based on all the above considerations, the effect of HVS masking characteristics is incorporated into the JND threshold value of sub-block $B_{u,v}$.

$$J'(u,v)=\omega(u,v)H(u,v)\cdot \tag{2}$$

At last, in order to improve the watermark invisibility and enhance the controllability on the embedded watermark strength, we scale J' to range $[a, b]$ as follows:

$$J(u,v)=\frac{b-a}{max(J')-min(J')}[J'(u,v)-min(J')]+a\cdot \tag{3}$$

where a and b are the predetermined threshold values, $J(u,v)$ is the JND threshold value of $B_{u,v}$. $max\{\cdot\}$ and $min\{\cdot\}$ denote the maximum and minimum of the matrix respectively.

Fig.1 illustrates the pre-calculated weight values of Lena, utilizing HVS. The brightness of the region is inversely proportional to the noise sensitivity of the region. Highly textured regions are shown in brighter shades of gray; the less textured areas, smooth areas and the prominent edge areas are depicted in darker shades of gray.

a b c

Fig. 1. Perceptual region classification of Lena image: a: Classification based on luminance, b: Classification based on entropy, c: Classification based on JND value

2.2 Watermark Embedding

In this paper, a meaningful binary image is used as a watermark. To prevent the watermark from tampering or unauthorized access by attackers, the watermark is first mapped into a pseudo-random data before insertion. The pseudo-random strategy is the same as that proposed in [2]. Let digital watermark W be a binary image of size $m \times n$, and PN be a binary pseudo-random matrix of size $m \times n$ generated by a secret key k. The binary image W and binary pseudo-random matrix PN are represented as $W=\{w(i,j)|\ w(i,j)\ \{0,1\}, 0 \le i \le m-1, 0 \le j \le n-1\}$, $PN=\{\ pn(i,j)|\ pn(i,j)\ \{0,1\}, 0 \le i \le m-1, 0 \le j \le n-1\ \}$.We adopt (4) to get the ultimate watermark W_1.

$$W_1 = \left\{ w_1(i,j) \middle| w_1(i,j) = w(i,j) \oplus pn(i,j), 0 \le i \le m-1, 0 \le j \le n-1 \right\}. \tag{4}$$

where denotes the exclusive OR.

The original image of size $M \times N$ is first divided into 4×4 blocks, and then each blocks are decomposed by 1-level DWT. The watermark is embedded into the original image by changing the polarity of the sum of the absolute value of coefficients of the vertical and the horizontal detail subbands of image sub-blocks.

Because the coefficients of the detail subbands of the image decomposed by DWT are consistent with the texture characteristics of original image, we can describe the watermark strength by the coefficients of the detail subbands. Let $\lambda^{\theta}_{u,v}(x,y)$ be the watermark strength of the coefficient (u,v) of detail subbands of sub-block $B_{u,v}$ on which 1-level DWT is applied, and it can be defined as (5).

$$\lambda^{\theta}_{u,v}(x,y) = \frac{\left| D^{\theta}_{u,v}(x,y) \right| + \Delta}{\sum_{x=0}^{1} \sum_{y=0}^{1} \left\{ \left| D^{\theta}_{u,v}(x,y) \right| + \Delta \right\}}. \tag{5}$$

where Δ s e predetermined threshold value whose purpose is to avoid that the denominator is zero. $\theta \in \{LH, HL\}$ (LH denotes vertical detail subband, HL denotes horizontal detail subband). $D^{\theta}_{u,v}$ denotes the detail subband of sub-block $B_{u,v}$. The more detailed steps are described in Fig.2.

if $w_1(u,v)==1, and \quad dif < J(u,v)$ *if* $D^{HL}_{u,v}(x,y)>0$ $\hat{D}^{HL}_{u,v}(x,y)=D^{HL}_{u,v}(x,y)+\lambda^{HL}_{u,v}(x,y)\frac{(J(u,v)-dif)}{2}$ *else* $\hat{D}^{HL}_{u,v}(x,y)=D^{HL}_{u,v}(x,y)-\lambda^{HL}_{u,v}(x,y)\frac{(J(u,v)-dif)}{2}$ *if* $D^{LH}_{u,v}(x,y)>0$ $D^{LH}_{u,v}(x,y)=D^{LH}_{u,v}(x,y)-\lambda^{LH}_{u,v}(x,y)\frac{(J(u,v)-dif)}{2}$ *else* $D^{LH}_{u,v}(x,y)=D^{LH}_{u,v}(x,y)+\lambda^{LH}_{u,v}(x,y)\frac{(J(u,v)-dif)}{2}$ *where* $dif=\sum_x\sum_y\left\|D^{HL}_{u,v}(x,y)\right\|-\sum_x\sum_y\left\|D^{LH}_{u,v}(x,y)\right\|$	*if* $w_1(u,v)==0, and \quad dif < J(u,v)$ *if* $D^{HL}_{u,v}(x,y)>0$ $\hat{D}^{HL}_{u,v}(x,y)=D^{HL}_{u,v}(x,y)-\lambda^{HL}_{u,v}(x,y)\frac{(J(u,v)-dif)}{2}$ *else* $\hat{D}^{HL}_{u,v}(x,y)=D^{HL}_{u,v}(x,y)+\lambda^{HL}_{u,v}(x,y)\frac{(J(u,v)-dif)}{2}$ *if* $D^{LH}_{u,v}(x,y)>0$ $D^{LH}_{u,v}(x,y)=D^{LH}_{u,v}(x,y)+\lambda^{LH}_{u,v}(x,y)\frac{(J(u,v)-dif)}{2}$ *else* $D^{LH}_{u,v}(x,y)=D^{LH}_{u,v}(x,y)-\lambda^{LH}_{u,v}(x,y)\frac{(J(u,v)-dif)}{2}$ *where* $dif=\sum_x\sum_y\left\|D^{LH}_{u,v}(x,y)\right\|-\sum_x\sum_y\left\|D^{HL}_{u,v}(x,y)\right\|$

Fig. 2. Watermarking edmbedding

This modification causes $\sum_x\sum_y\left|D^{HL}_{u,v}(x,y)\right|>\sum_x\sum_y\left|D^{LH}_{u,v}(x,y)\right|$ when "1" is embedded into the host image, otherwise, $\sum_x\sum_y\left|D^{HL}_{u,v}(x,y)\right|<\sum_x\sum_y\left|D^{LH}_{u,v}(x,y)\right|$.

Finally 1-level inverse DWT is applied to each 4×4 image sub-block to get the watermarked image.

2.3 Watermark Extraction

Watermark extraction is basically the inverse procedure of watermark embedding. Firstly, 1-level DWT is applied to the 4×4 sub-blocks of the watermarked image. Let $\tilde{D}^{\theta}_{u,v}$ be the detail subbands of sub-block $\tilde{B}_{u,v}$. Then the watermark extraction is depicted as follows:

$$\tilde{W}_1=\begin{cases}1, & if \quad \sum_x\sum_y\left|\tilde{D}^{HL}_{u,v}(x,y)\right|\geq\sum_x\sum_y\left|\tilde{D}^{LH}_{u,v}(x,y)\right| \\ 0, & if \quad \sum_x\sum_y\left|\tilde{D}^{HL}_{u,v}(x,y)\right|<\sum_x\sum_y\left|\tilde{D}^{LH}_{u,v}(x,y)\right|\end{cases}. \tag{6}$$

Finally, a binary pseudo-random matrix *PN* is generated with the same secret key in the watermark embedding. We can use (7) to get the ultimate extracted watermark $\tilde{W}$:

$$\tilde{W}=\left\{\tilde{w}(i,j)\middle|\tilde{w}(i,j)=\tilde{w}_1(i,j)\oplus pn(i,j), 0\leq i\leq m-1, 0\leq j\leq n-1\right\}. \tag{7}$$

2.4 Tamper Detection and Localization

The difference watermark image E is defined as follows:

$$E=\left\{e(i,j)\middle|e(i,j)=\left|w(i,j)-\tilde{w}(i,j)\right|,0\le i\le m-1,0\le j\le n-1\right\}. \tag{8}$$

The difference watermark image has value 1 (white pixel) if a watermarked 4×4 sub-block is falsely detected, 0 (black pixel) otherwise. Most of the watermark error pixels are spread on the difference watermark image or the extracted watermark when incidental attacks are made on the image. On the contrary, most of the watermark error pixels are concentrated in certain regions.

Based on the assumption, we can give a subjective evaluation of the image authenticity by the difference watermark image. At the same time, an objective quantitative method can be used to decide whether the tested image is maliciously tampered or innocuously altered.

For an error detection pixel in the difference watermark image, it is a tampered pixel if at least one of its eight neighbor pixels are error detection pixel, and a maliciously tampered pixel if at least two of its surrounding eight neighbors are error detection pixel, and isolated error detection pixels in the difference watermark image (see Fig.3) are considered as noise caused by error detection. We first define the following parameters:

$$\phi_{error}=\left\{\text{the total of error detection pixels in the difference watermark image}\right\}. \tag{9}$$

$$\phi_{tampered}=\left\{\text{the total of tampered pixels in the difference watermark image}\right\}. \tag{10}$$

$$\phi_{mal-tampered}=\left\{\text{the total of maliciously tampered pixels in the difference watermark image}\right\}. \tag{11}$$

$$\phi=\left\{\text{the total of pixels in the difference watermark image}\right\}. \tag{12}$$

$$\rho=\frac{\phi_{error}}{\phi}=\frac{\phi_{error}}{M\times N}. \tag{13}$$

$$\mu=\frac{\phi_{mal-tampered}}{E\left(\phi_{tampered}\right)}, \text{ where } E(\quad) \text{ denotes the mathematical expectation.} \tag{14}$$

Using the above parameters, we define the following rules to judge whether a modification is malicious or incidental:

A. If ρ=0 the tested image has been neither maliciously tampered nor incidental distorted.
B. If ρ>0 and $\mu<\zeta$, then the tested image has encountered only incidental distortions, where ζ is the preset threshold, $\zeta\in$ [0.5, 1].
C. If ρ>0 and $\mu\ge\zeta$, then the tested image has been maliciously tampered.

Moreover, for an inauthentic image, we can remove noise-like pixel and create a compact tamper region in the difference watermark image by applying mathematical morphology operations to the difference watermark image. Specific method is presented in Bani's scheme[10].

2.5 Threshold Selection

The embedding algorithm is designed in such a way that for an unwatermarked image the probability of the value of a pixel to be detected is 0 or 1 is 0.5. Thus, in the difference watermark image (Fig.3), the probability of a pixel with value 0 or 1 is 0.5, Let t_i denote that a pixel is a tampered pixel for an unwatermarked image, P_{t_i} denote the probability of t_i, similarly, Let mal_i denote that a pixel is a maliciously tampered pixel for an unwatermarked image, P_{mal_i} denote the probability of mal_i, then

$$\begin{aligned} P_{t_i} &= \frac{1}{2}\left\{1-\left(\frac{1}{2}\right)^8\right\} = \frac{255}{512}, \quad E(t_i) = \frac{255}{512}. \\ P_{mal_i} &= \frac{1}{2}\left\{1-\left(\frac{1}{2}\right)^8 - 8\times\left(\frac{1}{2}\right)^8\right\} = \frac{247}{512}, \quad E(mal_i) = \frac{247}{512}, \quad D(mal_i) = \frac{247}{512}\times\frac{265}{512}. \end{aligned} \tag{15}$$

where $E(\ )$, $D(\ )$ denotes the mathematical expectation and the variance, respectively.

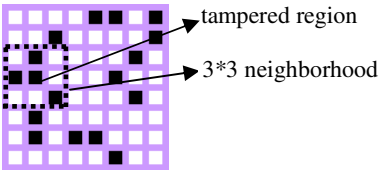


Fig. 3. The tampered region and the slide window in the difference watermark image

Because $\phi_{tampered}$, $\phi_{mal-tampered}$ obeys a binomial distribution, $\phi_{tampered}$, $\phi_{mal-tampered}$ approximately obeys a norm distribution plug in larger and larger values for $M\times N$, and then

$$\begin{aligned} E(\phi_{tampered}) &= E\left(\sum_{i=1}^{m\times n} t_i\right) = \frac{255}{512}\times m\times n. \\ E(\phi_{mal-tampered}) &= E\left(\sum_{i=1}^{m\times n} mal_i\right) = \frac{247}{512}\times m\times n, \quad D(\phi_{mal-tampered}) = D\left(\sum_{i=1}^{m\times n} mal_i\right) = \frac{247}{512}\times\frac{265}{512}\times m\times n. \end{aligned} \tag{16}$$

Let probability of false positive be 10^{-8}, then

$$\begin{aligned} & P(\phi_{mal-tampered} \ge \tau) = 1 - P(\phi_{mal-tampered} < \tau) \approx 1 - \Phi\left(\frac{\tau - 0.4824\times m\times n}{\sqrt{0.2497\times m\times n}}\right) \ge 1-10^{-8} \\ & \overset{m=n=128}{\Rightarrow} 1-\Phi\left(\frac{\tau - 0.4824\times 128\times 128}{\sqrt{0.2497\times 128\times 128}}\right) \ge 1-10^{-8} \overset{monotonic}{\Rightarrow} \tau \ge 7545.1 \Rightarrow \zeta \ge \frac{7545.1}{E(\phi_{tampered})} = 0.9246 \end{aligned} \tag{17}$$

So, we can consider that the attack on the watermarked image is malicious if μ *is* greater than $\zeta(0.9246)$, non-malicious otherwise.

3 Experimental Results

In this section, the simulated experiments are carried out to illustrate the effect of the proposed scheme. In experiments, we adopt a gray-scale image with size 512 512 as the original image, and use a bi-valued image with size 128 128 as the watermark image. Let β, I_1, I_2 in (1) be 0.60, 80, 180, respectively; a, b in (3) are set to 20, 50, respectively; Δ in (5) is 1.

3.1 Invisibility

To evaluate the invisibility of the watermark, we test the proposed algorithm on a series of standard images. Table 1 shows the PSNR values of the watermarked images. Take "Lena" image as an example, Fig.4 shows the experimental result on invisibility.

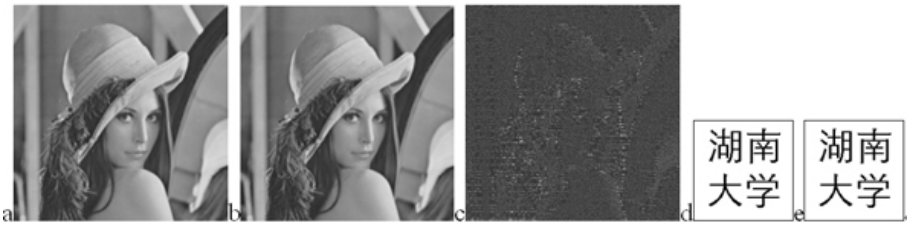


Fig. 4. Invisibility: a: Original image, b: Watermarked image, c: Absolute difference between the original image and the watermarked one, magnified by a factor 10, d: The original watermark, e: The extracted watermark

Table 1. PSNR values of the watermarked images (db)

lena	baboon	babara	boat	F16	water
40.28	35.80	36.57	37.98	36.86	38.93

From Table 1 and Fig.4, we can find that the watermarked images have high PSNR values, and the watermark is invisible to human eyes. At the same time, we notice that the embedding watermark is adaptive to the original image features by observing the difference images. Even any visual difference was hardly noticeable on the watermarked image with lower PSNR value, because of taking full advantage of HVS characteristics.

3.2 The Effectiveness of Classification Rule

In section 2.4, we defined a classification rule that is used to judge whether the tested image is malicious tampered or not. In this section, we will test the effectiveness of the classification rule under a series of attacks made on the watermarked content. Table 2 displays the experimental results. From Table 2, we can see that the attack on

the watermarked image is malicious if μ is greater than $\zeta(0.9246)$, non-malicious otherwise, which shows the effectiveness of the classification rule and the threshold selection.

Table 2. Classification of attacks

Attacks	ρ	μ	Malicious or non-malicious
JPEG(50)	0.2657	0.0189	non-malicious
JPEG(40)	0.2814	0.0206	non-malicious
JPEG(30)	0.2883	0.0239	non-malicious
low-pass filtering	0.2037	0.6674	non-malicious
Histogram equalization	0	0	no attack
5% salt & pepper noise	0.1490	0.4983	non-malicious
3% Gaussian noise	0.4131	0.7561	non-malicious
Rotation by 1°	0.4590	0.9783	malicious
Cropping in the upper-left quarter	0.1469	0.9859	malicious
Cropping in the upper-left 16×16 block	0.0005	0.9848	malicious

3.3 Robustness

This section examines the robustness against non-malicious attacks, including JPEG compression with quality factor 80,70,60,50,40, low-pass filtering, sharpening, Histogram equalization, 5% salt & pepper noise, 3% Gaussian noise. The corresponding extracted watermarks are shown in Fig.5a-j, the original image, the watermarked image and the original watermark are shown in Fig.4a,b,d, respectively. The performance of extracted watermarks are evaluated by normalized cross-correlation (NC)[9]. These results indicate that the proposed algorithm is robust against non-malicious operations.

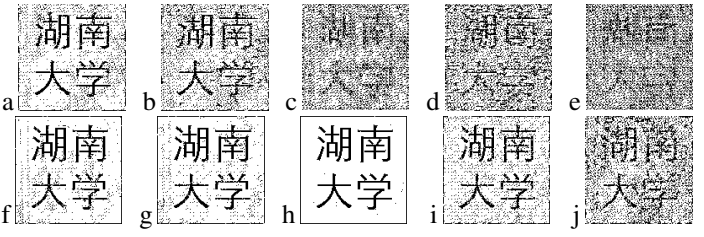


Fig. 5. Robustness against non-malicious attack. a: NC=0.9070, b: NC=0.8174, c: NC=0.7394, d: NC=0.6874, e: NC=0.6543, f: NC=0.9572, g: NC=0.9616, h: NC=0.999, i: NC=0.9077, j: NC=0.7370.

3.4 Fragility and the Localization Property

In this section, experiments are done on testing the fragility to malicious tamper and the ability to locate the tamper region under malicious tamper. The results are shown in Fig.6. In the tamper-detection image, the white and black areas are the tampered and non-tampered areas, respectively. These results indicate that the proposed scheme can afford a visual region for tamper, and locate accurately the tampered regions.

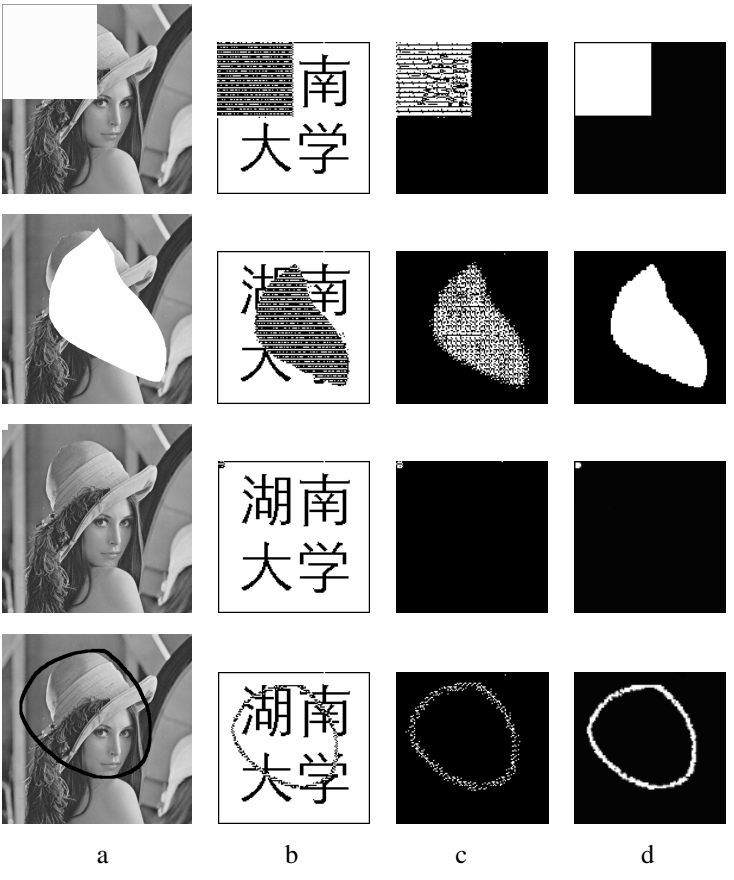


Fig. 6. Tamper detection and location. a: tampered image, b: the extracted watermark, c: difference watermark image, d: detected tampered region

4 Concluding Remarks

This paper describes an image-adaptive semi-fragile watermarking scheme for image authentication and tamper detection that operates in DWT domain. The experimental results show the effect of the proposed algorithm. The algorithm can distinguish effectively malicious attacks from non-malicious attacks on watermarked contents, and

even locate accurately the tampered blocks. The embedded watermark has good invisibility, fragility to malicious attacks and robustness against non-malicious attacks. Moreover, it is adaptive to the original image characteristics because of the utilization of HVS. Future work will involve further study for human visual system masking characteristics, and reversible fragile watermarking using HVS model.

Acknowledgements

This work is supported by National Natural Science Foundation of China (NSFC No. 60373062, 60573045), Hunan Provincial Young Research Project of Education Department of China (No. 04B016), Guangdong Provincial Research Project of Science and Technology (No. 2004B16001006).

References

1. Yu-jie, S., Tie-niu, T.: A Brief Review on Fragile Watermarking Based Image Authentication. Journal of image and graphics 8(A)1, 1–7 (2003)
2. Wu, X., Hu, J., Gu, Z., et al.: A Secure Semi-Fragile Watermarking for Image Authentication Based on Integer Wavelet Transform with Parameters. In: Proc. of Australasian Information Security Workshop 2005, Newcastle, Australia (2005)
3. Ishihara, N., Abe, K.: A Semi-Fragile Watermarking Scheme for Image Authentication and Tamper-Proofing. In: Proc. of 2nd International Conference on Intelligent Computing and Information Systems, Cairo, Egypt, pp. 590–595 (2005)
4. Lin, C.Y., Chang, S.F.: Semi-Fragile Watermarking for Authenticating JPEG Visual Content. SPIE Security and Watermarking of Multimedia Contents II, EI' 00, San Jose, CA, pp. 140–151(2000)
5. Feng, G.-r., Jiang, L.-g., He, C.: Permutation-Based Semi-Fragile Watermark Scheme. IEICE Trans. Fundamentals. E88-A(1), 374–377 (2005)
6. Lu, Z.-M., Liu, C.-H., Xu, D.-G., Sun, S.-H.: Semi-fragile Image Watermarking Method Based on Index Constrained Vector Quantisation. IEE Electronics Letters 39(1), 35–36 (2003)
7. Kang, H., Park, J.: A Semi-Fragile Watermarking Using JND. In: Proceedings of Pacific Rim Workshop on Digital Steganography 2003, Kitakyushu, Japan (2003)
8. Liu, Y., Gao, W., Yao, H., Liu, S.: A Texture-based Tamper Detection Scheme by Fragile Watermark. In: The 2004 IEEE International Symposium on Circuits and Systems, ISCAS 2004, Vancouver, CA, pp. 177–180 (May23-26, 2004)
9. Hengfu, Y., Zihua, Y., Mingfang, J.: Content Based Image Public Watermarking. In: proceedings of Eurographics Workshop on multimedia 2004, pp. 163–172 (2004)
10. Bartollini, F., Tefas, A., Barni, M., Pitas, I.: Image Authentication Techniques for Surveillance Applications. In: Proc. IEEE, vol. 89 (10), pp. 1403–1418. IEEE Computer Society Press, Los Alamitos (2001)

Identification of Fuzzy Set-Based Fuzzy Systems by Means of Data Granulation and Genetic Optimization

Keon-Jun Park[1], Sung-Kwun Oh[1], Hyun-Ki Kim[1], Witold Pedrycz[2], and Seong-Whan Jang[3]

[1] Department of Electrical Engineering, The University of Suwon, San 2-2 Wau-ri, Bongdam-eup, Hwaseong-si, Gyeonggi-do, 445-743, South Korea
ohsk@suwon.ac.kr
[2] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2G6, Canada and Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
[3] Department of Electrical Electronic and Information Engineering, Wonkwang University, 344-2, Shinyong-Dong, Iksan, Chon-Buk, 570-749, South Korea
swhjang@wonkwang.ac.kr

Abstract. This paper proposes an identification of fuzzy set-based fuzzy systems formed by using respective fuzzy spaces (fuzzy set). This model implements system structure and parameter identification by means of information granulation and genetic algorithms. Information granules are sought as associated collections of objects (data, in particular) drawn together by the criteria of proximity, similarity, or functionality. Information granulation realized with HCM clustering help determine the initial parameters of fuzzy model such as the initial apexes of the membership functions in the premise and the initial values of polynomial functions in the consequence. And the initial parameters are tuned by means of the genetic algorithms and the least square method. To optimally identify the structure and parameters we exploit the consecutive optimization of fuzzy set-based fuzzy model by means of genetic algorithms. An aggregate objective function is constructed in order to strike a sound balance between the approximation and generalization capabilities of the fuzzy model. The experimental part of the studies involves two representative numerical examples.

1 Introduction

Fuzzy set theory has been introduced to model uncertain and/or ambiguous characteristics inherent to experimental data. Since its inception, the research of fuzzy logic has been a focal point of various endeavors and demonstrated many fruitful results both in theory and application. Linguistic modeling [1] and fuzzy relation equation-based approach [2] were proposed as primordial identification methods for fuzzy models. The general class of Sugeno-Takagi models [3] gave rise to more sophisticated rule-based systems where the rules come with conclusions forming local regression models. While appealing with respect to the basic topology (a modular fuzzy model composed of a series of rules) [4], these models still await formal solutions as far as the structure optimization of the model is concerned, say a

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1076 – 1085, 2007.
© Springer-Verlag Berlin Heidelberg 2007

construction of the underlying fuzzy sets—information granules being viewed as basic building blocks of any fuzzy model. Some enhancements to the model have been proposed by Oh and Pedrycz [5], yet the problem of finding "good" initial parameters of the fuzzy sets in the rules remains open.

In this study we consider the problem of fuzzy set-based fuzzy model that is a development of information granules-fuzzy sets. The design methodology emerges as a hybrid structural and parametric optimization. We exploit the consecutive optimization of fuzzy model that is formed by using respective fuzzy spaces to optimally identify the structure and parameters. An aggregate objective function with some weighting factor is proposed so that we could achieve a sound balance between accuracy and generalization abilities of the fuzzy model. The proposed model is evaluated through numeric experimentation.

2 Information Granulation (IG)

Information granules [6], [7] are informally viewed as linked collections of objects (data point, in particular) drawn together by the criteria of proximity, similarity, or functionality. Granulation of information is an inherent and omnipresent activity of human beings carried out with intent of better understanding of the problem. In particular, granulation of information is aimed at splitting the problem into several manageable chunks. In this way, we partition this problem into a series of well-defined subproblems (modules) of a far lower computational complexity than the original one. The form of information granulation (IG) themselves becomes an important design feature of the fuzzy model, which are geared toward capturing relationships between information granules. Information granulation realized with HCM clustering help determine the initial parameters of fuzzy model such as the initial apexes of the membership functions in the premise and the initial values of polynomial functions in the consequence. It is worth emphasizing that the HCM clustering [8] has been used extensively not only to organize and categorize data, but it becomes useful in data compression and model identification.

3 Fuzzy Set-Based Fuzzy Systems by Means of IG

3.1 Premise Identification

The HCM clustering helps us organize the data into cluster, and in this way we take into account the characteristics of the experimental data. In the regions where some clusters of data have occurred, we end up with some fuzzy sets that help represent the specificity of the data set. In the sequel, the modal values of the clusters are refined using genetic optimization. The identification of the premise part of fuzzy rules that is formed by the isolated input space of each input variable is completed in the following manner.

Given is a set of data $\mathbf{U}=\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l ; \mathbf{y}\}$, where $\mathbf{x}_k =[x_{1k}, \ldots, x_{mk}]^T$, $\mathbf{y} =[y_1, \ldots, y_m]^T$, l is the number of variables and , m is the number of data.

[Step 1] Arrange a set of data **U** into data set $\mathbf{X}_k$.

$$\mathbf{X}_k=[\mathbf{x}_k\ ;\ \mathbf{y}] \tag{1}$$

[Step 2] Complete the HCM clustering to determine the centers (prototypes) $\mathbf{v}_{kg}$ with data set $\mathbf{X}_k$. and calculate the center vectors $\mathbf{v}_{kg}$ of each cluster.

$$\mathrm{v}_{kg}=\{v_{k1},\ v_{k2},\ \ldots,\ v_{kc}\} \tag{2}$$

Where, g = 1, 2, …, c.
[Step 3] Partition the corresponding isolated input space using the prototypes of the clusters $\mathbf{v}_{kg}$. Associate each clusters with some meaning, say Small, Big, etc.
[Step 4] Set the initial apexes of the membership functions using the prototypes $\mathbf{v}_{kg}$.

3.2 Consequence Identification

The characteristics of input-output data is also involved in the conclusion parts as follows:

[Step 1] Find a set of data included in the isolated fuzzy space of the j-th rule.
[Step 2] Compute the prototypes $\mathbf{V}_j$ of the data set by taking the arithmetic mean of each rule.

$$\mathrm{V}_j=\{V_{1j},V_{2j},\ldots,V_{kj};M_j\} \tag{3}$$

Where, j=1, 2, …, n. V_{kj} and M_j are prototypes of input and output data, respectively.
[Step 3] Set the initial values of polynomial functions with the center vectors $\mathbf{V}_j$.

The identification of the conclusion parts of the rules deals with a selection of their structure that is followed by the determination of the respective parameters of the local functions occurring there.

In case of Type 2: linear Inference (linear conclusion)
The conclusion is expressed in the form of a linear relationship between inputs and output variable. The rules come in the following form

$$R^j: If\ x_k\ is\ A_{kc}\ then\ y_j-M_j=f_j(x_1,\cdots,x_k) \tag{4}$$

The calculations of the numeric output of the model, based on the activation (matching) levels of the rules there, are carried out in the well-known format

$$y^*=\frac{\sum_{j=1}^{n}w_{ji}y_i}{\sum_{j=1}^{n}w_{ji}}=\frac{\sum_{j=1}^{n}w_{ji}(f_j(x_1,\cdots,x_k)+M_j)}{\sum_{j=1}^{n}w_{ji}}=\sum_{j=1}^{n}\hat{w}_{ji}(a_{j0}+a_{j1}(x_1-V_{j1})+\cdots+a_{jk}(x_k-V_{jk})+M_j) \tag{5}$$

where R^j is the j-th fuzzy rule, x_k represents the input variables, A_{kc} is a membership function of fuzzy sets, a_{j0} is a constant, n is the number of fuzzy rules, y^* is the inferred output value, w_{ji} is the premise fitness matching R^j (activation level).

Here we use two performance indexes as the mean squared error (MSE) and root mean squared error (RMSE).

$$\mathrm{PI} = \frac{1}{m}\sum_{i=1}^{m}(y_i - y_i^*)^2 \quad , \quad \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - y_i^*)^2} \tag{6}$$

The minimal value produced by the least-squares method is governed by the following expression:

$$\hat{\mathbf{a}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{7}$$

Where,

$\mathbf{x}_i^T = [\,\hat{w}_{1i} \cdots \hat{w}_{ni} \;\; (x_{1i} - V_{11})\hat{w}_{1i} \cdots (x_{1i} - V_{1n})\hat{w}_{ni} \cdots (x_{ki} - V_{k1})\hat{w}_{1i} \cdots (x_{ki} - V_{kn})\hat{w}_{ni}\,]$,

$\hat{\mathbf{a}} = [a_{10} \cdots a_{n0}\; a_{11} \cdots a_{n1} \cdots a_{1k} \cdots a_{nk}]^T$, $\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_i \quad \cdots \quad \mathbf{x}_m]^T$

$$\mathbf{Y} = \left[\, y_1 - \left(\sum_{j=1}^{n} M_j w_{j1}\right) \quad y_2 - \left(\sum_{j=1}^{n} M_j w_{j2}\right) \quad \cdots \quad y_m - \left(\sum_{j=1}^{n} M_j w_{jm}\right)\right]^T .$$

4 Consecutive Optimization by Means of Genetic Algorithms

4.1 Genetic Algorithms

The optimization of fuzzy models may not be fully supported by the standard gradient-based optimization techniques, because of the nonlinearity of fuzzy models represented by rules based on linguistic levels. This forces us to explore other optimization techniques such as genetic algorithms. It has been demonstrated that genetic algorithms [9] are useful in a global optimization of such problems given their ability to efficiently use historical information to produce new improved solutions with enhanced performance.

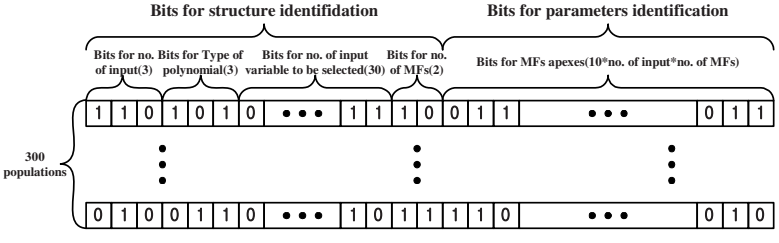


Fig. 1. The consecutive coding to identify the structure and parameters

To optimally identify the fuzzy model we determine such an initial structure as the number of input variables, input variables being selected and the number of membership functions standing in the premise and the type of polynomial in conclusion. The membership parameters of the premise are genetically optimized. Figure 1 shows an arrangement of the content of the string to be used in genetic optimization to consecutively identify the fuzzy model. Here, parentheses mean the number of chromosomes for each parameter.

4.2 Objective Function with Weighting Factor

The objective function (performance index) [5] is a basic mechanism guiding the evolutionary search carried out in the solution space. The objective function includes both the training data and testing data and comes as a convex combination of the two components.

$$f(PI,\ E_PI) = \theta \times PI + (1 - \theta) \times E_PI \tag{8}$$

Here, *PI* and *E_PI* denote the performance index for the training data and testing (validation) data, respectively. θ is a weighting factor that allows us to form a sound balance between the performance of the model for the training and testing data. Depending upon the values of the weighting factor, several specific cases of the objective function are worth distinguishing.

(i) If $\theta = 1$ then the model is optimized based on the training data. No testing data is taken into consideration.
(ii) If $\theta = 0.5$ then both the training and testing data are taken into account. Moreover it is assumed that they exhibit the same impact on the performance of the model.
(iii) The case $\theta = \alpha$ where $\alpha \in [0, 1]$ embraces both the cases stated above. The choice of α establishes a certain tradeoff between the approximation and generalization aspects of the fuzzy model.

In the ensuing numerical experiment, we consider the performance index (6), while θ is regarded to be an extra parameter.

5 Experimental Studies

5.1 Chaotic Mackey-Glass Time Series

In this section we consider comprehensive numeric studies illustrating the design of the fuzzy model. We demonstrate how IG-based FIS can be utilized to predict future values of a chaotic time series. The performance of the proposed model is also contrasted with some other models existing in the literature. The time series is generated by the chaotic Mackey–Glass differential delay equation [10] of the form:

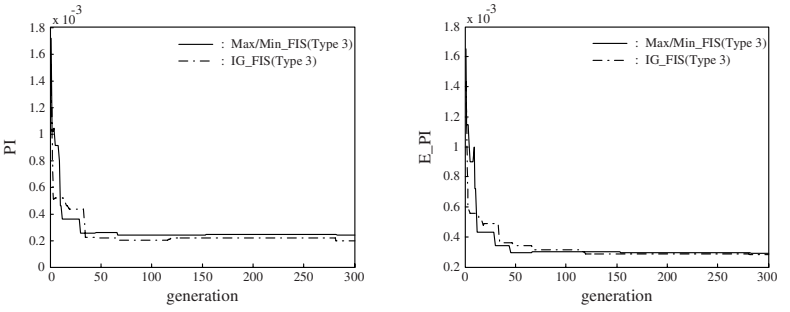
$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{9}$$

The prediction of future values of this series arises is a benchmark problem that has been used and reported by a number of researchers. From the Mackey–Glass time series $x(t)$, we extracted 1000 input–output data pairs for the type from the following the type of vector format such as: [$x(t\text{-}30)$, $x(t\text{-}24)$, $x(t\text{-}18)$, $x(t\text{-}12)$, $x(t\text{-}6)$, $x(t)$; $x(t+6)$] where t = 118–1117. The first 500 pairs were used as the training data set while the remaining 500 pairs were the testing data set for assessing the predictive performance. We consider the MSE being regarded here as a performance index. We carried out the structure and parameters identification on a basis of the experimental data using GAs

to design Max_Min-based and IG-based fuzzy model. For each fuzzy model, we consecutively conducted the optimization of the parameters of the premise membership functions. Table 1 summarizes the performance index for Max_Min-based and IG-based fuzzy model. It shows that the performance of the IG-based fuzzy is better than that of a Max_Min-based fuzzy model.

Table 1. Performance index of Max_Min-based and IG-based fuzzy model

Model	θ	Input variable	No. Of MFs	Type	PI	E_PI
Max/Min_FIS	0.0	$x(t\text{-}30)$, $x(t\text{-}18)$, $x(t\text{-}12)$, $x(t)$	5+5+5+5	Type 3	0.00022	0.00028
	0.25				0.00022	0.00026
	0.5				0.00024	0.00029
	0.75				0.00023	0.00029
	1.0				0.00023	0.00038
IG_FIS	0.0	$x(t\text{-}30)$, $x(t\text{-}18)$, $x(t\text{-}12)$, $x(t)$	5+5+5+5	Type 3	0.00023	0.00027
	0.25				0.00023	0.00028
	0.5				0.00019	0.00028
	0.75				0.00022	0.00028
	1.0				0.00021	0.00036

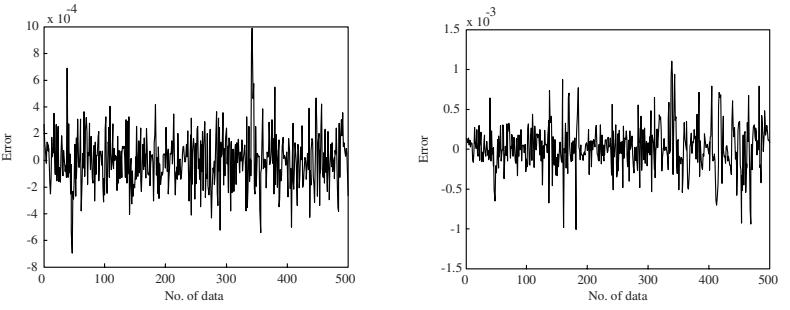


(a) PI (b) E_PI

Fig. 2. Convergence process of performance index for Max_Min-based and IG-based fuzzy model with input variables, $x(t\text{-}30)$, $x(t\text{-}18)$, $x(t\text{-}12)$, $x(t)$ of type 3 and θ=0.5

Figure 2 depicts the values of the performance index produced in successive generation of the genetic optimization for Max_Min-based and IG-based fuzzy model with input variables, $x(t\text{-}30)$, $x(t\text{-}18)$, $x(t\text{-}12)$, $x(t)$ of type 3 and θ=0.5. The resulting diction error for the training and testing data for IG-based fuzzy model with input variables, $x(t\text{-}30)$, $x(t\text{-}18)$, $x(t\text{-}12)$, $x(t)$ of type 3 and θ=0.5 are displayed in Figure 3.

The identification error (performance index) of the proposed model is also compared with the performance of some other models; refer to Table 2. Here the non-dimensional error index (NDEI) is defined as the root mean square errors divided by the standard deviation of the target series.



(a) Error of training data (b) Error of testing data

Fig. 3. Error curve of training and testing data for IG-based fuzzy model with input variables, $x(t\text{-}30)$, $x(t\text{-}18)$, $x(t\text{-}12)$, $x(t)$ of type 3 and θ=0.5

Table 2. Comparison of identification error with previous models

Model	No. of rules	PI_t	PI	E_PI	NDEI
Wang's model [11]	7	0.004			
	23	0.013			
	31	0.010			
Cascaded-correlation NN [12]					0.06
Backpropagation MLP [12]					0.02
6^{th}-order polynomial [12]					0.04
ANFIS [13]	16		0.0016	0.0015	0.007
FNN model [14]			0.014	0.009	
Recurrent neural network [15]		0.0138			
Our model	20		0.0001	0.0002	0.0012

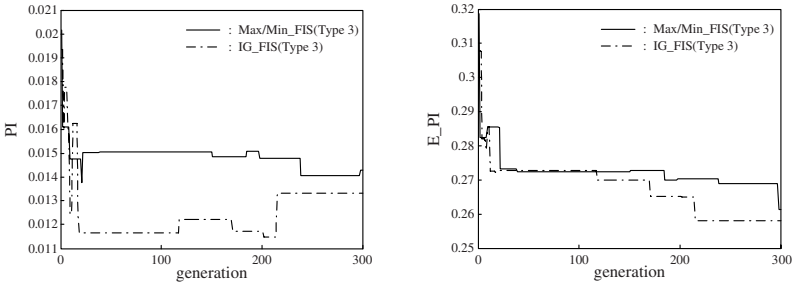
5.2 Gas Furnace Process

We illustrate the performance of the model and elaborate on its development by experimenting with data coming from the gas furnace process. The time series data (296 input-output pairs) resulting from the gas furnace process has been intensively studied in the previous literatures [5,18,19]. The delayed terms of methane gas flow rate $u(t)$ and carbon dioxide density $y(t)$ are used as six input variables with vector formats such as [$u(t\text{-}3)$, $u(t\text{-}2)$, $u(t\text{-}1)$, $y(t\text{-}3)$, $y(t\text{-}2)$, y(t-1)]. And as output variable $y(t)$ is used. The first one (consisting of 148 pairs) was used for training. The remaining part of the series serves as a testing set. We consider the MSE being regarded here as a performance index. We carried out the structure and parameters identification on a basis of the experimental data using GAs to design Max_Min-based and IG-based fuzzy model. For each fuzzy model, we consecutively conducted the optimization of the parameters of the premise membership functions.

Table 3 summarizes the performance index for Max_Min-based and IG-based fuzzy model. From the Table 3 we know that the performance of IG-based fuzzy model is better than that of the Max_Min-based fuzzy model.

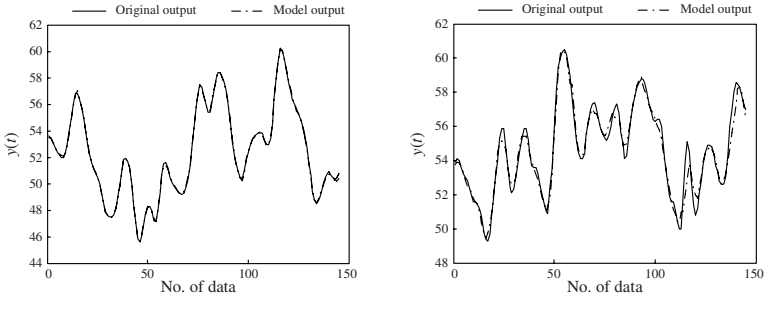
Table 3. Performance index of Max_Min-based and IG-based fuzzy model

Model	θ	Input variable	No. Of MFs	Type	PI	E_PI
Max/Min_FIS	0.0	$y(t\text{-}2)$, $y(t\text{-}1)$	3+3	Type 1	0.181	0.190
	0.25	$y(t\text{-}2)$, $y(t\text{-}1)$	4+4	Type 4	0.086	0.198
	0.5	$y(t\text{-}2)$, $y(t\text{-}1)$	5+5	Type 4	0.078	0.204
	0.75	$u(t\text{-}3)$, $y(t\text{-}1)$	4+4	Type 3	0.014	0.261
	1.0	$u(t\text{-}3)$, $y(t\text{-}1)$	5+5	Type 3	0.011	0.417
IG_FIS	0.0	$y(t\text{-}2)$, $y(t\text{-}1)$	4+4	Type 4	0.083	0.198
	0.25	$y(t\text{-}2)$, $y(t\text{-}1)$	5+5	Type 2	0.076	0.194
	0.5	$y(t\text{-}2)$, $y(t\text{-}1)$	5+5	Type 4	0.076	0.194
	0.75	$u(t\text{-}3)$, $y(t\text{-}1)$	5+5	Type 3	0.013	0.258
	1.0	$u(t\text{-}3)$, $y(t\text{-}1)$	5+5	Type 3	0.010	0.291



(a) PI (b) E_PI

Fig. 4. Convergence process of performance index for Max_Min-based and IG-based fuzzy model with input variables, $u(t\text{-}3)$, $(t\text{-}1)$ of type 3 and θ=0.75



(a) Training data (b) Testing data

Fig. 5. The model output of training and testing data for IG-based fuzzy with input variables, $u(t\text{-}3)$, $(t\text{-}1)$ of type 3 and θ=0.75

Figure 4 shows the values of the performance index produced in successive generation of the genetic optimization for Max_Min-based and IG-based fuzzy model with input variables, $u(t\text{-}3)$, $(t\text{-}1)$ of type 3 and θ=0.75. It is obvious that the performance of an IG-based fuzzy model is good due to the characteristics of input-output data. The model output for training and testing data for IG-based fuzzy model with input variables, $u(t\text{-}3)$, $(t\text{-}1)$ of type 3 and θ=0.75 is displayed in figure 5. The identification error (performance index) of the proposed model is also compared to the performance of some other models in Table 4. The performance of the proposed model is better in the sense of its approximation and prediction abilities than other works studied in the literatures as shown in Table 4.

Table 4. Comparison of identification error with previous models

Model		PI_t	PI	E_PI	No. of rules
Tong's model[16]		0.469			19
Pedrycz's model[2]		0.776			20
Xu's model[17]		0.328			25
Sugeno's model[4]		0.355			6
Oh et al.'s Model[5,18]	Simplified		0.024	0.328	4
	Linear		0.022	0.326	4
			0.021	0.364	6
HCM+GA[19]	Simplified		0.035	0.289	4
			0.022	0.333	6
	Linear		0.026	0.272	4
			0.020	0.264	6
Our model			0.013	0.258	10

6 Conclusions

The identification of fuzzy set-based fuzzy systems by means of information granulation and consecutive optimization of the structure and parameters is studied in this paper. This model used the isolated input space for each input variable and defined the fuzzy space by information granule. Information granulation realized with HCM clustering help determine the initial parameters of fuzzy model such as the initial apexes of the membership functions in the premise and the initial values of polynomial function in the consequence. The membership parameters are tuned (adjusted) with the aid of the genetic algorithms and the least square method. Through the use of the certain form of the performance we were able to achieve a balance between the approximation and generalization abilities of the resulting model. Two examples are given to illustrate the validity and compare its performance with those of other models.

Acknowledgements. This work was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MOST)(F01-2007-000-10058-0)

References

1. Tong, R.M.: Synthesis of fuzzy models for industrial processes. Int. J Gen Syst. 4, 143–162 (1978)
2. Pedrycz, W.: An identification algorithm in fuzzy relational system. Fuzzy Sets Syst. 13, 153–167 (1984)
3. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst, Cybern. SMC 15(1), 116–132 (1985)
4. Sugeno, M., Yasukawa, T.: Linguistic modeling based on numerical data. In: IFSA'91 Brussels, Computer, Management & System Science, pp. 264–267 (1991)
5. Oh, S.K., Pedrycz, W.: Identification of fuzzy systems by means of an auto-tuning algorithm and its application to nonlinear systems. Fuzzy Sets and Syst. 115(2), 205–230 (2000)
6. Zadeh, L.A.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Syst. 90, 111–117 (1997)
7. Pderycz, W., Vukovich, G.: Granular neural networks. Neurocomputing. 36, 205–224 (2001)
8. Krishnaiah, P.R., Kanal, L.N. (eds.): Classification, pattern recognition, and reduction of dimensionality. Handbook of Statistics, vol. 2. North-Holland, Amsterdam (1982)
9. Golderg, D.E.: Genetic Algorithm in search, Optimization & Machine Learning. Addison-Wesley, Reading (1989)
10. Mackey, M.C., Glass, L.: Oscillation and chaos in physiological control systems. Science. 197, 287–289 (1977)
11. Wang, L.X., Mendel, J.M.: Generating fuzzy rules from numerical data with applications. IEEE Trans. Systems, Man, Cybern. 22(6), 1414–1427 (1992)
12. Crowder III, R.S.: Predicting the Mackey-Glass time series with cascade-correlation learning. In: Touretzky, D., Hinton, G., Sejnowski, T. (eds.) Proceedings of the 1990 Connectionist Models Summer School, pp. 117–123. Carnegic Mellon University (1990)
13. Jang, J.R.: ANFIS: Adaptive-Network-Based Fuzzy Inference System. IEEE Trans. System, Man, and Cybern. 23(3), 665–685 (1993)
14. Maguire, L.P., Roche, B., McGinnity, T.M., McDaid, L.J.: Predicting a chaotic time series using a fuzzy neural network. Information Sciences. 112, 125–136 (1998)
15. Li, C.J., Huang, T.Y.: Automatic structure and parameter training methods for modeling of mechanical systems by recurrent neural networks. Applied Mathematical Modeling. 23, 933–944 (1999)
16. Tong, R.M.: evaluation of fuzzy models derived from experimental data. Fuzzy Sets Syst. 13, 1–12 (1980)
17. Xu, C.W., Zailu, Y.: Fuzzy model identification self-learning for dynamic system. IEEE Trans. on Syst. Man, Cybern., SMC 17(4), 683–689 (1987)
18. Park, C.S., Oh, S.K., Pedrycz, W.: Fuzzy Identification by means of Auto-Tuning Algorithm and Weighting Factor. The Third Asian Fuzzy Systems Symposium(AFSS), pp. 701–706 (1998)
19. Park, B.J., Pedrycz, W., Oh, S.-K.: Identification of Fuzzy Models with the Aid of Evolutionary Data Granulation. IEE Proc.-Control Theory and Applications 148(5), 406–418 (2001)
20. Park, H.S., Oh, S.K.: Fuzzy Relation-based Fuzzy Neural-Networks Using a Hybrid Identification Algorithm. Int. J. of Control, Automations, and Systems 1(3), 289–300 (2003)

Public Key Encryption with Keyword Search Based on K-Resilient IBE

Dalia Khader

University of Bath, Department of Computer Science,
ddk20@bath.ac.uk

Abstract. An encrypted email is sent from Bob to Alice. A gateway wants to check whether a certain keyword exists in an email or not for some reason (e.g. routing). Nevertheless Alice does not want the email to be decrypted by anyone except her including the gateway itself. This is a scenario where public key encryption with keyword search (PEKS) is needed. In this paper we construct a new scheme (KR-PEKS) the K-Resilient Public Key Encryption with Keyword Search. The new scheme is secure under a chosen keyword attack without the random oracle. The ability of constructing a Public Key Encryption with Keyword Search from an Identity Based Encryption was used in the construction of the KR-PEKS. The security of the new scheme was proved by showing that the used IBE has a notion of key privacy. The scheme was then modified in two different ways in order to fulfill each of the following: the first modification was done to enable multiple keyword search and the other was done to remove the need of secure channels.

1 Introduction

Bob wants to send Alice confidential emails and in order to ensure that no one except her can read it, he encrypts the emails before sending them so that Alice, and her alone, will possess the capability of decrypting it. Consider the scenario where Alice would like to download only the urgent emails to her mobile, leaving the rest to check later from her computer. Alice requires access to the email server (gateway) so that she can search her emails for the keyword "urgent" prior to downloading them. The server should facilitate the search for this keyword without being able to decrypt Alice's private emails.

This scenario was first introduced in Boneh et al.'s paper [4]. They presented a general scheme called PEKS where Alice gives trapdoors for the words she wants the gateway to search for. The trapdoors come in the form of some kind of data that is used to test the existence of keywords within an email without revealing any other information(Section 3).

In [4] the authors constructed several schemes based on different security models but these schemes either had some limitation on the number of keywords to search for or were not secure enough (ie. were proven secure in random oracle).

In [2] the authors pointed out two important features that were not covered in [4]. The first one was the ability to search for multiple keywords. The second

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1086–1095, 2007.
© Springer-Verlag Berlin Heidelberg 2007

characteristic, put forwards in Section 3.1 of this paper, was elimination of the requirement of secure channels, for sending trapdoors. These two new features issues are explained in details in the paper and new schemes to facilitate them will be introduced in this report paper(Section 3.6 and 3.7).

The ability of constructing IBE from PEKS was explored in [4]. IBE is a public key encryption where the public key is a direct product of the identity of the user [12] [5]. Building a PEKS from an IBE needs the latter to have some extra properties such as the notion of key privacy. Key privacy is a security property that implies that an adversary should not be able to guess which ID from a set of ID's was used in encrypting some email(Section 3.3).

Heng and Kurosawain [10] proposed a scheme called K-resilient IBE, this scheme does not lead to breach of privacy [1]. The basic K-resilient IBE was used to construct a PEKS that is fully secure (Section 3.4).

The next Section entails some important concepts that form the foundation for our work. Following that, Section 3 explains the PEKS scheme, its security notions, the relation with IBE, and last but not least the construction of a new PEKS scheme using the K-Resilient IBE was introduced. The last Section in this paper concludes the results of our work.

2 Preliminaries

In this Section we will go through some definitions that will be used further in this document.

2.1 Decisional Diffie-Hellman

The Decisional Diffie Hellman (DDH) Problem [7] is the ability to distinguish between $\langle g, g^a, g^b, g^{ab}\rangle$ and $\langle g, g^a, g^b, T\rangle$ where $a, b, c \in Z_q$,$g \in G_q$, G_q is a group of prime order q, and T is a random element that belongs to G_q.

The quadruple $\langle g, g^a, g^b, g^{ab}\rangle$ is called the real quadruple and the quadruple $\langle g, g^a, g^b, T\rangle$ is called the random quadruple. So if we have an adversary D that takes $X, Y, T \in G_q$ and returns a bit $d \in \{0, 1\}$, consider the following two experiments.

For both experiments $X \leftarrow g^x$; $Y \leftarrow g^y$

- $Exp^{ddh-real}_{G_q,D}$
  $T \leftarrow g^{xy}$
  $d \leftarrow D(q, g, X, Y, T)$
  Return d

- $Exp^{ddh-rand}_{G_q,D}$
  A random $T \in G_q$
  $d \leftarrow D(q, g, X, Y, T)$
  Return d

The advantage of the adversary D in solving the Diffie-Hellman is defined as follows:-

$$Adv^{ddh}_{G_q,D} = Pr[Exp^{ddh-real}_{G_q,D} = 1] - Pr[Exp^{ddh-rand}_{G_q,D} = 1]$$

If this advantage is negligible for any adversary D then we say DDH is hard to solve. The DDH was used in PEKS Section 3.4.

2.2 Hash Functions

A family of hash functions $\mathcal{H} = (G; H)$ is defined by two algorithms [9]. G is a probabilistic generator algorithm that takes the security parameter k as input and returns a key K. H is a deterministic evaluation algorithm that takes the key K and a string $M \in \{0,1\}^*$ and returns a string $H_k(M) \in \{0,1\}^{k-1}$.

Definition. Let $\mathcal{H} = (G; H)$ be a family of hash functions and let C be an adversary [9]. We consider the following experiment:

Experiment $Exp^{cr}_{\mathcal{H},C}(k)$

$K \leftarrow G(k); (x_0; x_1) \leftarrow C(K)$

if $((x_0 \neq x_1) \wedge (H_K(x_0) = H_K(x_1)))$ then return 1 else return 0.

We define the advantage of C via $Adv^{cr}_{\mathcal{H},C}(k) = Pr[Exp^{cr}_{\mathcal{H},C}(k) = 1]$:
The family of hash functions $\mathcal{H}$ is collision-resistant if the advantage of C is negligible for every algorithm C whose time-complexity is polynomial in k.

3 Public Key Encryption with Keyword Search

An encrypted email is sent from Bob to Alice [4]. The gateway wants to check whether a certain keyword exists in an email or not for some reason (for example routing). Nevertheless Alice does not want the email to be decrypted by anyone except her, not even the gateway itself. This is a scenario where public key encryption with keyword search (PEKS) is needed. PEKS encrypts the keywords in a different manner than the rest of the email. The gateway is given "trapdoors" corresponding to particular keywords. Using the PEKS of a word and trapdoor of a keyword, the gateway can test whether the encrypted word is the particular keyword or not.

General scheme. According to [4] a PEKS consists of four algorithms as described below :

- `KeyGen(`s`)`: Take a security parameter s and generate two keys a public key A_{pub} and private key A_{priv}
- `PEKS(`A_{pub}, W`)`: It produces a searchable encryption for a keyword W using a public key A_{pub}
- `Trapdoor(`A_{priv}, W`)`: Produce a trapdoor for a certain word using the private key.
- `Test(`A_{pub}, S, T_w`)`: Given the public key A_{pub}, some searchable encryption S where $S = PEKS(A_{pub}, W')$, and the trapdoor T_w to a keyword W. Determine whether or not the word we are looking for W and the word encrypted W' are equal.

So Bob sends Alice through the gateway the following:

$$[E(A_{pub}, M), PEKS(A_{pub}, W_1), PEKS(A_{pub}, W_2), ..., PEKS(A_{pub}, W_m)]$$

where $PEKS(A_{pub}, W_i)$ is a searchable encryption of the keywords and $E(A_{pub}, M)$ is a standard public key encryption of the rest of the message M.

3.1 Security Notions Related to PEKS

Security under a Chosen Keyword Attack (CKA). For a PEKS to be considered secure we need to guarantee that no information about a keyword is revealed unless the trapdoor of that word is available [4]. To define security against an active adversary A we use the following game between A and challenger.

- `CKA-Setup`: The challenger runs the key generation algorithm and gives the A_{pub} to adversary A and keeps A_{priv} to itself.
- `CKA-Phase 1`: A asks the challenger for trapdoors corresponding to keywords of its choice.
- `CKA-Challenge`: The adversary decides when phase 1 ends. Then it chooses two words W_0, W_1 to be challenged on. The two words should not be among those for which A obtained a trapdoor in phase 1. The challenger picks a random bit $b \in \{0, 1\}$ and gives attacker. $C = PEKS(A_{pub}, W_b)$.
- `CKA-Phase 2`: A asks for more trapdoors like in phase 1 for any word of its choice except for the W_0, W_1.
- `CKA-Guess`: A outputs its guess of b' and if $b' = b$ that means A guessed the encrypted message and the adversary wins.

We say that the scheme is secure against a chosen keyword attack (CKA) if A has a low advantage of guessing the right word being encrypted.

Secure Channels: In the PEKS scheme in [4] there is a need to have a secure channel between Alice and the server, so that an eavesdropper (Eve) can not get hold of the trapdoors sent. No one but the server should be capable of testing emails for certain keywords. This is one of the drawback that the authors of [2] tried to solve by generating a public and a private key that belong to the server. The PEKS algorithm was modified to encrypt keywords using both Alice's and the server's public key, while the testing algorithm needs the server's private key as an input. In this way the scheme is secure channel free (SCF-PEKS) because Eve can not obtain the server's private key, therefore can not test.

The SCF-PEKS is said to be IND-SCF-CKA secure when it ensures that the server that has obtained the trapdoors for given keywords cannot tell a PEKS ciphertext is the result of encrypting which keyword, and an outsider adversary that did not get the server's private key cannot distinguish the PEKS ciphertexts, even if it gets all the trapdoors for the keywords that it queries

3.2 Handling Multiple Keywords

Multiple Keyword search in a PEKS is the capability of searching for more than one word either disjunctively or conjunctively. In PEKS [4] the only way to do this is to search for each word separately and then do the disjunctive or conjunctive operations on the result of the testing algorithm. This technique is impractical when it comes to a large number of keywords in one conjunctive search request, because every email is searched for every single keyword. In [8]

a new scheme was suggested for conjunctive search called PECK. The scheme substitutes the PEKS algorithm with a PECK algorithm that encrypts a query of keywords. The testing is done with a trapdoor for each query instead of each word. So Bob sends Alice the following:

$$[E(A_{pub}, M), PECK(A_{pub}, (W_1, W_2, ..., W_m))]$$

We say that the scheme is secure against a chosen keyword attack (CKA) if an adversary has a low advantage in guessing the right query of keywords being encrypted.

3.3 The Strong Relation Between IBE and PEKS

In [4] the authors showed how the algorithms used in IBE can be used for constructing a PEKS. They showed how with the four algorithms in an IBE Setup, Extract, Encrypt, and Decrypt [12] [5] could be used for to achieve the purpose of a PEKS scheme. So if keyword was used in place of an ID in the IBE scheme. The Setup algorithm will be equivalent to the KeyGen algorithm in a PEKS. Extracting the private key in the IBE will be in replace of generating trapdoors for keywords in PEKS scheme. Now if the Encryption algorithm in the IBE was used to encrypt some zero string of a certain length, the result will be a PEKS ciphertext that could later be tested by using the Decryption algorithm in an IBE and checking whether the result is the same string of zeros. The problem is that the ciphertext could expose the public key (W) used to create it. So we need to derive a notion of key privacy [3] for IBE to ensure that the PEKS is secure under a chosen keyword attack. Key privacy is a security notion first introduced in [3]. If an adversary can not guess which ID of a set of ID's in an IBE scheme was used in encrypting a particular ciphertext then that IBE scheme does not lead to breach the privacy of its keys. This could be under chosen plaintext attack(IK-CPA) or chosen ciphertext attack(IK-CCA).

3.4 Construction of a PEKS from the K-Resilient IBE (KRPEKS)

Since the K-resilient IBE scheme suggested in [10] is said to be IK-CCA secure ([1] for proof) it was tempting to construct a PEKS using that scheme. As any other PEKS scheme there are four algorithms, summarized in the following.

- `KRPEKS-KeyGen`
  Step 1: Choose a group G of order q and two generators g_1, g_2
  Step 2: Choose 6 random k degree polynomials where the polynomials are chosen over Z_q
  $P_1(x) = d_0 + d_1x + d_2x^2 + ... + d_kx^k$; $P_2(x) = d'_0 + d'_1x + d'_2x^2 + ... + d'_kx^k$
  $F_1(x) = a_0 + a_1x + a_2x^2 + ... + a_kx^k$; $F_2(x) = a'_0 + a'_1x + a'_2x^2 + ... + a'_kx^k$
  $h_1(x) = b_0 + b_1x + b_2x^2 + ... + b_kx^k$; $h_2(x) = b'_0 + b'_1x + b'_2x^2 + ... + b'_kx^k$
  Step 3: For $0 \leq t \leq k$; Compute $A_t = g_1^{a_t} g_2^{a'_t}, B_t = g_1^{b_t} g_2^{b'_t}, D_t = g_1^{d_t} g_2^{d'_t}$
  Step 4: Choose a random collision resistant hash function H (Section 2.2)
  Step 5: Choose a random collision resistant hash function H' (Section 2.2)

Step 6: Assign $A_{priv} = \langle F_1, F_2, h_1, h_2, P_1, P_2 \rangle$
$A_{pub} = \langle g_1, g_2, A_0, ..., A_k, B_0, ..., B_k, D_0, ..., D_k, H, H' \rangle$

– `KRPEKS`
Step 1: Choose a random $r_1 \in Z_q$
Step 2: Compute $u_1 = g_1^{r_1}$; $u_2 = g_2^{r_1}$
Step 3: Calculate for each keyword w
$A_w \leftarrow \Pi_{t=0}^{k} A_t^{w^t}$; $B_w \leftarrow \Pi_{t=0}^{k} B_t^{w^t}$; $D_w \leftarrow \Pi_{t=0}^{k} D_t^{w^t}$
Step 4: $s \leftarrow D_w^{r_1}$
Step 5: Using the -exclusive or- operation calculate $e \leftarrow (0^k) \otimes H'(s)$
Step 6: $\alpha \leftarrow H(u_1, u_2, e)$
Step 7: $v_w \leftarrow (A_w)^{r_1}.(B_w)^{r_1 \alpha}$
Step 8: $C \leftarrow \langle u_1, u_2, e, v_w \rangle$

– `KRPEKS-Trapdoor`
Run Extract of the IBE and the output is the trapdoor
$T_w = \langle F_1(w), F_2(w), h_1(w), h_2(w), P_1(w), P_2(w) \rangle$.

– `KRPEKS-Test`
Step 1: $\alpha \leftarrow H(u_1, u_2, e)$
Step 2: Test if $v_w \neq (u_1)^{F_1(w)+h_1(w)\alpha}.(u_2)^{F_2(w)+h_2(w)\alpha}$
then Halt else go to Step 3.
Step 3: $s \leftarrow (u_1)^{P_1(w)}.(u_2)^{P_2(w)}$
Step 4: $m \leftarrow e \otimes H'(s)$
Step 5: If the resulting plaintext is a 0^k conclude
that C is an encryption of w.

The security of this scheme relies on DDH and the collision resistant hash function as shown in the next Section 3.5.

3.5 Security of KRPEKS Against CKA

The K-Resilient IBE scheme [10] is an identity based encryption that is based on the Decisional Diffie Hellman problem (DDH). The security of such scheme is based on the difficulty of solving DDH and whether the hash functions used are collision resistant or not. In [1], the following theorem was proved.

Theorem. Let G be a group of prime order q. If DDH is hard in G then KRIBE is said to be IK-CCA secure. So for any adversary A attacking the anonymity of KRIBE under a chosen ciphertext attack and making in total a $q_d(.)$ decryption oracle queries, there exist a distinguisher D_A for DDH and an adversary C attacking the collision resistance of H such that

$$Adv_{KRIBE,A}^{IK-CCA}(K) \leq 2Adv_{G,D_A}^{DDH}(K) + 2Adv_{H,C}^{CR}(K) + (q_d(K) + 2)/(2^{k-3})$$

Since KRIBE is IK-CCA secure [1]. Therefore if an adversary knows two IDs ID_0, ID_1 and is given a ciphertext encrypted using one of the IDs. The adversary would not to be able to guess which one was used unless the DDH is not hard or the hash function is not collision resistant. In this Section we show that since the PEKS was built from the KRIBE and KRIBE has key privacy notions then PEKS should logically be proved to be secure under a CKA.

Now if we compare both schemes the KRIBE in [10] and the KRPEKS we would notice that the key generation algorithm in the latter is the same as the setup in the former. The trapdoor is created the same way a secret key is created for an ID in the IBE scheme but instead of the IDs we have words. The encryption of IBE is equivalent to it for the PEKS but instead of a message, a k length zero string 0^k is encrypted. The testing of the existence of a keyword is done by using the same decryption algorithm of the IBE and checking whether the result is equal to 0^k. In other words if an adversary can not tell the difference between which ID was used to encrypt a given ciphertext. Then the same adversary would not know which word was used in creating ciphertext $C = PEKS(A_{pub}, W_b)$. The only difference between the two security notions is that instead of having a decryption oracle in proving IK-CCA in KRIBE we have a trapdoor oracle in the new PEKS. So someone can conclude that the advantage of guessing the right word depends on the DDH problem, the collision resistance of the hash function and Q_t where Q_t is the maximum number of trapdoor queries issued by the adversary.

3.6 Constructing K-Resilient SCF-PEKS Scheme

In [2] the authors constructed a SCF-PEKS using the same methodology used in the PEKS in [4]. In this Section we will try to build a SCF-PEKS using the KR-PEKS described in 3.4.

- $SCF - KRPEKS - CPG$ (Common Parameter Generator) :
  Step 1: Choose a group G and two generators g_1, g_2
  Step 2: Choose random k
  Step 3: Choose a random collision resistant hash function H (Section 2.2).
  Step 4: Calculate the common parameter $cp = \langle G, g_1, g_2, H, k\rangle$
- $SCF - KRPEKS - SKG(cp)$ (Server Key Generator) :
  Step 1: Choose 6 random k degree polynomials, chosen over Z_q
  $P_1(x) = d_0 + d_1x + d_2x^2 + ... + d_kx^k$; $P_2(x) = d'_0 + d'_1x + d'_2x^2 + ... + d'_kx^k$
  $F_1(x) = a_0 + a_1x + a_2x^2 + ... + a_kx^k$; $F_2(x) = a'_0 + a'_1x + a'_2x^2 + ... + a'_kx^k$
  $h_1(x) = b_0 + b_1x + b_2x^2 + ... + b_kx^k$; $h_2(x) = b'_0 + b'_1x + b'_2x^2 + ... + b'_kx^k$
  Step 2: For $0 \le t \le K$; Compute $A_t = g_1^{a_t} g_2^{a'_t}, B_t = g_1^{b_t} g_2^{b'_t}, D_t = g_1^{d_t} g_2^{d'_t}$
  Step 3: Assign $A_{priv_s} = \langle F_1, F_2, h_1, h_2, P_1, P_2\rangle$
  $A_{pub_s} = \langle g_1, g_2, A_0, ..., A_k, B_0, ..., B_k, D_0, ..., D_k, H\rangle$
- $SCF - KRPEKS - RKG(cp)$ (Reciever Key Generator) :
  Step 1: Choose 6 random k degree polynomials, chosen over Z_q
  $\hat{P_1}(x) = \hat{d_0} + \hat{d_1}x + \hat{d_2}x^2 + ... + \hat{d_k}x^k$; $\hat{P_2}(x) = \hat{d'_0} + \hat{d'_1}x + \hat{d'_2}x^2 + ... + \hat{d'_k}x^k$
  $\hat{F_1}(x) = \hat{a_0} + \hat{a_1}x + \hat{a_2}x^2 + ... + \hat{a_k}x^k$; $\hat{F_2}(x) = \hat{a'_0} + \hat{a'_1}x + \hat{a'_2}x^2 + ... + \hat{a'_k}x^k$
  $\hat{h_1}(x) = \hat{b_0} + \hat{b_1}x + \hat{b_2}x^2 + ... + \hat{b_k}x^k$; $\hat{h_2}(x) = \hat{b'_0} + \hat{b'_1}x + \hat{b'_2}x^2 + ... + \hat{b'_k}x^k$
  Step 2: For $0 \le t \le K$; Compute $\hat{A_t} = g_1^{\hat{a_t}} g_2^{\hat{a'_t}}, \hat{B_t} = g_1^{\hat{b_t}} g_2^{\hat{b'_t}}, \hat{D_t} = g_1^{\hat{d_t}} g_2^{\hat{d'_t}}$
  Step 3: Choose a random collision resistant hash function H' (Section 2.2).
  Step 4: Assign $A_{priv_r} = \langle \hat{F_1}, \hat{F_2}, \hat{h_1}, \hat{h_2}, \hat{P_1}, \hat{P_2}\rangle$
  $A_{pub_r} = \langle g_1, g_2, \hat{A_0}, ..., \hat{A_k}, \hat{B_0}, ..., \hat{B_k}, \hat{D_0}, ..., \hat{D_k}, H, H'\rangle$

– $SCF - KRPEKS$
Step 1: Choose a random $r_1 \in Z_q$
Step 2: Compute $u_1 = g_1^{r_1}$; $u_2 = g_2^{r_1}$
Step 3: Calculate $A_w \leftarrow \Pi_{t=0}^{k} A_t^{w^t}$; $B_w \leftarrow \Pi_{t=0}^{k} B_t^{w^t}$; $D_w \leftarrow \Pi_{t=0}^{k} D_t^{w^t}$
$\hat{A_w} \leftarrow \Pi_{t=0}^{k} \hat{A_t}^{w^t}$; $\hat{B_w} \leftarrow \Pi_{t=0}^{k} \hat{B_t}^{w^t}$; $\hat{D_w} \leftarrow \Pi_{t=0}^{k} \hat{D_t}^{w^t}$
Step 4: $s \leftarrow D_w^{r_1} \hat{D_w}^{r_1}$
Step 5: $e \leftarrow (0^k) \otimes H'(s)$
Step 6: $\alpha \leftarrow H(u_1, u_2, e)$
Step 7: $v_w \leftarrow ((A_w)(\hat{A_w}))^{r_1}.((B_w)(\hat{B_w}))^{r_1 \alpha}$
Step 8: $C \leftarrow \langle u_1, u_2, e, v_w \rangle$
– $SCF - KRPEKS - TG$ (Trapdoor Generator) :
Calculate: $T_w = \langle \hat{F_1}(W), \hat{F_2}(W), \hat{h_1}(W), \hat{h_2}(W), \hat{P_1}(W), \hat{P_2}(W) \rangle$
– $SCF - KRPEKS - T$ (Testing Algorithm) :
Step 1: $\alpha \leftarrow H(u_1, u_2, e)$
Step 2: Test if $v_w \neq (u_1)^{F_1(w)+h_1(w)\alpha+\hat{F_1}(w)+\hat{h_1}(w)\alpha}.(u_2)^{F_2(w)+h_2(w)\alpha+\hat{F_2}(w)+\hat{h_2}(w)\alpha}$
then halt else go to next step
Step 3: $s \leftarrow (u_1)^{P_1(w)+\hat{P_1}(w)}.(u_2)^{P_2(w)+\hat{P_2}(w)}$
Step 4: $m \leftarrow e \otimes H'(s)$
Step 5: If the resulting plaintext is a 0^k conclude
that C is the encryption of w.

Notice that the testing part can not be done except by the server. Therefore, the trapdoor could be sent via public channels.

3.7 Constructing a K-Resilient PECK

In [8] [11] the authors constructed a PECK by adopting ideas from the PEKS in [4]. In this paper we will try to build a PECK scheme by adopting ideas from the KR-PEKS. The four algorithms that form this scheme are described as follows.

– `KRPECK-KeyGen:`
Step 1: Choose a group G of order q and two generators g_1, g_2
Step 2: Choose 6 random k degree polynomials, chosen over Z_q
$P_1(x) = d_0 + d_1 x + d_2 x^2 + ... + d_k x^k$; $P_2(x) = d'_0 + d'_1 x + d'_2 x^2 + ... + d'_k x^k$
$F_1(x) = a_0 + a_1 x + a_2 x^2 + ... + a_k x^k$; $F_2(x) = a'_0 + a'_1 x + a'_2 x^2 + ... + a'_k x^k$
$h_1(x) = b_0 + b_1 x + b_2 x^2 + ... + b_k x^k$; $h_2(x) = b'_0 + b'_1 x + b'_2 x^2 + ... + b'_k x^k$
Step 2: For $0 \leq t \leq k$; Compute $A_t = g_1^{a_t} g_2^{a'_t}, B_t = g_1^{b_t} g_2^{b'_t}, D_t = g_1^{d_t} g_2^{d'_t}$
Step 3: Choose two random numbers $s_0, s_1 \in Z_q$
Step 4: Calculate $S = g_1^{s_0}.g_2^{s_1}$
Step 5: Choose a random collision resistant hash function H (Section 2.2).
Step 6: Calculate: $A_{priv} = \langle F_1, F_2, h_1, h_2, P_1, P_2, s_0, s_1 \rangle$
$A_{pub} = \langle g_1, g_2, A_0, ..., A_k, B_0, ..., B_k, D_0, ..., D_k, S, H \rangle$
– `KRPECK:`
Step 1: Choose a random $r_1 \in Z_q$
Step 2: Compute $u_1 = g_1^{r_1}$; $u_2 = g_2^{r_1}$

Step 3: Calculate for every W_i where $1 \leq i \leq m$

$$A_{w_i} \leftarrow \Pi_{t=0}^{k} A_t^{w_i^t} \; ; \; B_{w_i} \leftarrow \Pi_{t=0}^{k} B_t^{w_i^t} \; ; \; D_{w_i} \leftarrow \Pi_{t=0}^{k} D_t^{w_i^t}$$

Step 4: Calculate e_i where $1 \leq i \leq m$ and $e_i \leftarrow D_{w_i}^{r_1}$

Step 5: Calculate α_i where $1 \leq i \leq m$ and $\alpha_i \leftarrow H(u_1, u_2, e_i)$

Step 6: $v_{w_i} \leftarrow (A_{w_i})^{r_1}.(B_{w_i})^{r_1 \alpha_i}$

Step 7: $C \leftarrow \langle u_1, u_2, e_1, ..., e_m, v_{w_1}, ..., v_{w_m}, S^{r_1} \rangle$

– `KRPECK-Trapdoors:`

Step 1: Choose $\Omega_1, ..., \Omega_t$ where t is the number of keywords you want to search for and Ω_i is the keyword in position I_i

Step 2: $T_1 = P_1(\Omega_1) + P_1(\Omega_2) + ... + P_1(\Omega_t) + s_0$

Step 3: $T_2 = P_2(\Omega_1) + P_2(\Omega_2) + ... + P_2(\Omega_t) + s_1$

Step 4: For $1 \leq j \leq t$ Compute $\alpha_j \leftarrow H(u_1, u_2, e_{I_j})$

Step 5: $T_3 = F_1(\Omega_1) + ... + F_1(\Omega_t) + h_1(\Omega_1)\alpha_1 + ... + h_1(\Omega_t)\alpha_t$

Step 6: $T_4 = F_2(\Omega_1) + ... + F_2(\Omega_t) + h_2(\Omega_1)\alpha_1 + ... + h_2(\Omega_t)\alpha_t$

Step 7: $T_Q = \langle T_1, T_2, T_3, T_4, I_1, ..., I_t \rangle$

– `KRPECK-Test:`

Step 1: Test if $v_{w_{I_1}}.v_{w_{I_2}}...v_{w_{I_t}} \neq u_1^{T_3}.u_2^{T_4}$ then halt else do Step 2

Step 2: If $S^{r_1}.e_{I_1}.e_{I_2}.e_{I_3}...e_{I_t} = u_1^{T_1}.u_2^{T_2}$

then output "True" otherwise output false

If all the words exist, then definitely the condition of the if-statement in Step 2 in the testing algorithm will be true.

4 Conclusion

The main aim of this research was to have a PEKS that is secure under a standard model rather than the random oracle model only. To do so, the first step was finding an IBE scheme that has key privacy notions. Use of IBE with Weil pairing to build a PEKS was demonstrated in [5] and the scheme was secure under a chosen keyword attack but under the random oracle only. The IBE suggested by Boneh and Boyenin in [6] also was not useful in constructing a PEKS as shown in [2]. It was tempting to try to prove the K-resilient IBE [1] to have a notion of key privacy because it was shown in [3] that the Cramer-Shoup encryption is secure. The KRIBE adopted a lot of techniques from this encryption scheme and KRIBE was proved to be secure [1].

The new PEKS scheme was then used to construct a public key encryption with conjunctive keyword search and a public key encryption that does not need a secure channel.

However, the new PEKS scheme still has some drawbacks because of the limitations of the KRIBE scheme itself, where the number of malicious users is restricted to some value K. That is the number of trapdoors generated in the PEKS is limited to at most K. Nevertheless, that is not a serious problem where we could use a reasonably large K for email searching applications.

An additional concern lies in the basic formulation of the PEKS system. The idea of the sender, Bob having the sole power to decide which words to consider

as keywords for the recipient, Alice, may not be as convenient in reality. In fact, Alice should have all influence on the email sorting and one solution would be for her to cache a set of criteria in form of queries. In that way, the emails categorized as 'urgent' would have greater possibility of being what she considers imperative to read.

References

1. Public key encryption with keyword search based on k-resilient ibe (full version)
2. Baek, J., Naini, R., Susilo, W.: Public key encryption with keyword search revisited. Cryptology ePrint Archive, Report 2005/191 (2005), http://eprint.iacr.org/
3. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key cryptography. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 566–582. Springer, Heidelberg (2001)
4. Boneh, D., Crescenzo, G., Ostrovsky, R., Persiano, G.: Public-key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, Springer, Heidelberg (2004), citeseer.ist.psu.edu/boneh04public.html
5. Boneh, D., Franklin, M.: Identity based encryption from the Weil pairing. SIAM Journal on Computing 32(3), 586–615 (2003)
6. Boneh, D., Boyen, X.: Short signatures without random oracles. Cryptology ePrint Archive, Report 2004/171 (2004) http://eprint.iacr.org/
7. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
8. Cha, J., Park, D., Lee, P.: Searchable keyword-based encryption. Cryptology ePrint Archive, Report 2005/367 (2005) http://eprint.iacr.org/
9. Hayashi, R., Tanaka, K.: Elgamal and cramer shoup variants with anonymity using different groups extended abstract. C-200 (2004) http://www.is.titech.ac.jp/research/research-report/C/
10. Heng, S., Kurosawa, K.: K-resilient identity-based encryption in the standard model. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 67–80. Springer, Heidelberg (2004)
11. Park, D., Kim, K., Lee, P.: Public key encryption with conjunctive field keyword search. In: Lim, C.H., Yung, M. (eds.) WISA 2004. LNCS, vol. 3325, pp. 73–86. Springer, Heidelberg (2005)
12. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Parikh, R. (ed.) Logics of Programs. LNCS, vol. 193, pp. 47–53. Springer, Heidelberg (1985)

Efficient Partially Blind Signatures with Provable Security

Qianhong Wu[1], Willy Susilo[1], Yi Mu[1], and Fanguo Zhang[2]

[1] School of Information Technology and Computer Science
University of Wollongong
Wollongong NSW 2522, Australia
{qhw,wsusilo,ymu}@uow.edu.au
[2] School of Information Science and Technology
Sun Yat-sen University, Guangzhou 510275
Guangdong Province, P. R. China
isdzhfg@zsu.edu.cn

Abstract. Blind signatures play a central role in applications such as e-cash and e-voting systems. The notion of partially blind signature is a more applicable variant such that the part of the message contains some common information pre-agreed by the signer and the signature requester in an unblinded form. In this paper, we propose two efficient partially blind signatures with provable security in the random oracle model. The former is based on witness indistinguishable (WI) signatures. Compared with the state-of-the-art construction due to Abe and Fujisaki [1], our scheme is 25% more efficient while enjoys the same level of security. The latter is a partially blind Schnorr signature without relying on witness indistinguishability. It enjoys the same level of security and efficiency as the underlying blind signature.

1 Introduction

The notion of blind signature was first introduced by Chaum in [4]. After Chaum's first scheme based on RSA, some discrete-log based signature schemes were converted into blind signatures (e.g., [6], [8], [10]). Chaum remarked that the notion is the only way to implement electronic cash simulating the ones in reality. However, when the original notion of blind signatures is implemented for e-cash systems, as the bank knows nothing about the resulting signature or the signed message, some problems are posed, such as how to prevent misuse of the signature or how to embed the information such as the issuing date, expiration date, face value of e-cash and so on. There are two conventional solutions. The first one is that the bank uses different public keys to link with such common information. In this case, the shops and customers must always carry a list of those public keys in their electronic wallet, which is typically a smart card whose memory is very limited. The second solution is that the bank can use the cut-and-choose algorithm [4] in the withdrawal phase. This solution is also very inefficient.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1096–1105, 2007.
© Springer-Verlag Berlin Heidelberg 2007

To address such issues in applications, partially blind signatures were introduced by Abe and Fujisaki [1] to allow the signer to explicitly include some pre-agreed information in blind signatures. Using partially blind signatures in e-cash systems, the bank can be relieved from maintaining a unlimitedly growing database. The bank assures that each e-cash contains the information it desires, such as the date information. By embedding an expiration date into each e-cash issued by the bank, all expired e-cash recorded in the banks database can be removed. At the same time, since face values are embedded into e-cash, the bank knows the value on each e-cash blindly issued. The notion attracts a lot of attentions and has been extensively implemented under different assumptions (e.g., [1], [2], [7], [18], [5], [21]).

In this paper, we propose two efficient partially blind signatures with provable security in the random oracle model. The former is based on witness indistinguishable signatures [3]. Compared with the state-of-the-art construction due to Abe and Fujisaki [1], our scheme is 25% more efficient while enjoys the same level of security. The latter is a partially blind Schnorr signature without relying on witness indistinguishability. The merit is that, after an efficient publicly available evolution of public keys, we achieve partially blindness from blind signatures *without* introducing additional overhead or degrading the security of the underlying schemes.

2 Notations and Definitions

2.1 Notations

A negligible function is a function $\varepsilon(\lambda)$ such that for all polynomials $poly(\lambda)$, $1/\varepsilon(\lambda) < 1/poly(\lambda)$ holds for all sufficient large λ. PPT stands for *probabilistic polynomial-time*. $KS\{x : y = f(x)\}(m)$ represents a knowledge signature on message m that is transformed from the zero-knowledge proof of a secret value x satisfying $y = f(x)$ with the well-known Fiat-Shamir transformation. An efficient algorithm $\mathcal{A}(\cdot)$ is a probabilistic Turing machine running in expected polynomial time. An adversary $\mathcal{A}$ is a PPT interactive Turing machine. If $\mathcal{A}(\cdot)$ is an efficient algorithm and x is an input for $\mathcal{A}$, then $\mathcal{A}(x)$ denotes the probability space that assigns to a string σ the probability that $\mathcal{A}$, on input x, outputs σ. An efficient algorithm is deterministic if for every input x, the probability mass of $\mathcal{A}(x)$ is concentrated on a signed output string σ. For a probability space $\mathbb{P}$, $x \leftarrow \mathbb{P}$ denotes the algorithm that samples a random element according to $\mathbb{P}$. For a finite set $\mathbb{X}$, $x \leftarrow \mathbb{X}$ denotes the algorithm that samples an element uniformly at random from $\mathbb{X}$.

2.2 Partially Blind Signatures

We review the security definitions of partially blind signatures in [2] which follows from [9]. They are similar to those of conventional blind signature.

A partially blind signature is a tuple $PBS=(\mathcal{G}^{PBS}(\cdot), Sig^{PBS}(\cdot), Ver^{PBS}(\cdot))$:

- $(sk, pk) \leftarrow \mathcal{G}^{PBS}(\mathbf{1}^{\lambda})$ is a PPT algorithm which, on input a security parameter λ, outputs the signer's private/public key pair (sk, pk).
- $\sigma \leftarrow Sig^{PBS}_{sk;m}(pk, \texttt{info})$ is a two-party protocol between a signer $\mathcal{S}$ and a user $\mathcal{U}$. They have common input pk, which is the public key, and a pre-agreed common information `info` between the signer and the user. $\mathcal{S}$ has private input sk, which is its secret key, and its inner random coin flips. $\mathcal{U}$ has private input m, which is the message to be signed, and its inner random coin flips. At the end of the interaction between the two parties, $\mathcal{S}$ outputs one of the two messages: *completed*, *not − completed*, and $\mathcal{U}$ outputs either $fail$ or a signature σ on m.
- $0/1 \leftarrow Ver^{PBS}(m, \texttt{info}, \sigma, pk)$ is a polynomial-time deterministic algorithm which, on input a message-signature pair (m, σ), pre-agreed information `info` and the signer's public key pk, returns 1 or 0 for *accept* or *reject*, respectively. If *accept*, the message-signature pair is valid.

2.3 Security of Partially Blind Signatures

Similarly to blind signatures, the security of partially blind signature schemes has three critical aspects: correctness, partial blindness and $(l, l+1)$-unforgeability. They are formally defined as follows.

Definition 1. (Correctness.) *If the signer $\mathcal{S}$ and the user $\mathcal{U}$ honestly follow the protocol $Sig^{PBS}(\cdot)$, $\mathcal{U}$ will output a signature σ on m accepted by $Ver^{PBS}(\cdot)$ with a dominant probability.*

Definition 2. (Partial Blindness.) *The partial blindness is defined via an experiment involving an adversarial signer $\mathcal{A}$. The experiment is parameterized by a bit b and security parameter λ. First $\mathcal{G}^{PBS}(1^{\lambda})$ is correctly run to generate the adversarial signer $\mathcal{A}$'s secret key/public key pair (sk, pk). Then, $\mathcal{A}$ outputs a pair of messages (m_0, m_1) lexicographically ordered. In the next stage of the experiment $\mathcal{A}$ engages in two (possibly correlated and interleaved) runs with two honest users, with inputs m_b and m_{1-b}, respectively. If both users obtain valid signatures, on their respective message, $\mathcal{A}$ is also given these two signatures; otherwise there is no extra input to $\mathcal{A}$; in either case, $\mathcal{A}$ is required to output a bit b'. The advantage of $\mathcal{A}$ is defined by:*

$$Adv^{Bld}_{PBS,\mathcal{A}}(\lambda) = 2\Pr[b = b'] - 1.$$

A PBS scheme satisfies partial blindness, if for any PPT adversary signer $\mathcal{A}$, the function $Adv^{Bld}_{PBS,Ad}(\lambda)$ is negligible in λ. If $\mathcal{A}$'s computational power is unlimited, the blindness is unconditional.

Definition 3. ($(l,l+1)$-Unforgeability.) *The $(l,l+1)$-Unforgeability of a partially blind signature scheme is defined via an experiment parameterized by security parameters l and λ. The experiment involves an adversarial user $\mathcal{A}$ and is as follows: First $\mathcal{G}^{PBS}(1^{\lambda})$ is run to generate the signer's secret/public key pair (sk, pk). The signer and the adversarial user $\mathcal{A}$ also negotiate with the common information `info`. Then, the adversary user $\mathcal{A}$ engages in polynomially many*

(possibly correlated and interleaved) runs of the protocol with the signer. Finally $\mathcal{A}$ *outputs a list of message-signature pairs* $\{(m_1, \sigma_1), (m_2, \sigma_2), ..., (m_t, \sigma_t)\}$ *with* $m_i \neq m_j$ *and a string* `info'`. *Let* l *be the number of runs successfully completed by the signer. Define the advantage of* $\mathcal{A}$

$$Adv^{Unf}_{PBS,\mathcal{A}}(\lambda) = \Pr[\forall 1 \leq i \leq t, Ver^{PBS}(m_i, \texttt{info}, \sigma_i, pk) = 1 \wedge (l < t)] + \Pr[\exists 1 \leq i \leq t, Ver^{PBS}(m_i, \texttt{info}', \sigma_i, pk) = 1 \wedge \texttt{info}' \neq \texttt{info}]$$

and say that the PBS scheme is $(l, l+1)$*-unforgeable if* $Adv^{Unf}_{PBS,\mathcal{A}}(\lambda)$ *is negligible for any PPT adversary user* $\mathcal{A}$.

3 Proposed Partially Blind Signatures

In this section, we propose two partially blind signatures. The first one is a blind version of the knowledge proof signature that the signer knows a discrete logarithm of Schnorr public key y or the hashed value of common information c to a given base g. So we name it *partially blind OR-Schnorr signature*. It is similar to the construction in [2] but more efficient. The second one is a blind version of knowledge proof signature where the signer knows $(x + F(c))^{-1}$ which is the inverse of the sum of Schnorr secret key and the hashed value of common information c. Hence, we refer to it as partially blind inverse-Schnorr signature. It can be generalized to suit other signatures derived from knowledge proofs of discrete logarithms.

3.1 Partially Blind OR-Schnorr Signature

Let G be a cyclic group with prime order q, and g an element in G whose order is q. We assume that any polynomial-time algorithm solves $\log_g h$ in $\mathbb{Z}_q$ only with negligible probability when h is selected randomly from G. Let $H : \{0,1\}^* \to \mathbb{Z}_q$ and $F : \{0,1\}^* \to G$ be public cryptographic hash functions. Let $x \in \mathbb{Z}_q$ be a secret key and $y = g^x$ be the corresponding public key. The signer and the user first agree on common information c in a predetermined way. Then, they execute the signature issuing protocol on the user's blind message m as follows.

- (Initialization.) The signer randomly selects $r, u \leftarrow \mathbb{Z}_q^*$, and computes $z = F(c)$, $a = g^r z^u$. The signer sends a to the user as a commitment.
- (Blinding.) The user randomly selects $t_1, t_2, t_3 \leftarrow \mathbb{Z}_q$ as blind factors, and computes $z = F(c), \alpha = a g^{t_1} y^{t_2} z^{t_3}$, $\varepsilon = H(\alpha||c||z||m)$, $e = \varepsilon - t_2 - t_3 \bmod q$. The user sends e to the signer.
- (Signing.) The signer sends back v, s to the user, where $v = e - u \bmod q$, $s = r - vx \bmod q$.
- (Unblinding.) The user computes $\sigma = s + t_1 \bmod q$, $\rho = v + t_2 \bmod q$, $\delta = e - v + t_3 \bmod q$. It outputs (σ, ρ, δ) as the resulting signature on the message m and the pre-agreed common information c .
- (Verification.) The signature is valid if and only if $\rho + \delta = H(g^\sigma y^\rho z^\delta||c||z||m)$ and $z = F(c)$.

Now we consider the security of the scheme. Note that, in the Signing step, the signer returns $v = e - u$ and the user knows e. So the user can compute u;

that is, the user learns the component u of the indistinguishable witness (r, u) such that $a = g^r z^u$. However, due to the witness indistinguishability, u can be replaced by any other value u' in the random oracle model. This is indeed the key point that the signer procedure can be simulated without knowledge of the secret key x satisfying $y = g^x$ in the random oracle model. The detailed simulation is provided in Theorem 3.

Theorem 1. *(Correctness) If the signer and user follows the protocol, the output of the user will be accepted by the verification algorithm.*

Proof. Note that
$g^\sigma y^\rho z^\delta = g^{s+t_1} y^{v+t_2} z^{e-v+t_3} = g^s y^v z^{e-v} g^{t_1} y^{t_2} z^{t_3} =$
$g^{r-vx} y^v z^u g^{t_1} y^{t_2} z^{t_3} = a g^{t_1} y^{t_2} z^{t_3} = \alpha$ and $\rho + \delta = u + v + t_2 + t_3 = e + t_2 + t_3 = \varepsilon$. It follows that $\rho + \delta = H(g^\sigma y^\rho z^\delta || c || z || m)$, $z = F(c)$. The verification holds.

Theorem 2. *(Partial Blindness) The above partially blind signature is unconditionally blind.*

Proof. It is sufficient to prove that, for any view (a, e, v, s, c) of the adversary signer $\mathcal{S}^*$ and any signature pair $(\sigma, \rho, \delta, m, c)$, there exists a blind factor tuple that maps the view and the signature pair.

For $i = 0, 1$, let (a_i, e_i, v_i, s_i, c) be views of $\mathcal{S}^*$ and $(\sigma_j, \rho_j, \delta_j, m_j, c)$ two valid partially blind signatures from user $j = 0, 1$, respectively. Let $t_1 = \sigma_j - s_i, t_2 = \rho_j - v_i, t_3 = \delta_j - e_i + v_i$. Since $a_i = g^{r_i} z^{u_i}$ and the signatures are valid, it follows that

$\rho_j + \delta_j = H(g^{\sigma_j} y^{\rho_j} z^{\delta_j} || z || m_j)$
$= H(a_i g^{-r_i} z^{-u_i} g^{\sigma_j} y^{\rho_j} z^{\delta_j} || z || m_j) = H(a_i g^{\sigma_j - r_i} z^{\delta_j - u_i} y^{\rho_j} || z || m_j)$
$= H(a_i g^{\sigma_j - s_i - v_i x} z^{\delta_j - e_i + v_i} y^{\rho_j} || z || m_j) = H(a_i g^{\sigma_j - s_i} z^{\delta_j - e_i + v_i} y^{\rho_j - v_i} || z || m_j)$
$= H(a_i g^{t_1} y^{t_2} z^{t_3} || z || m_j)$.

Hence, given $j \in \{0, 1\}$, (a_0, e_0, v_0, s_0, c) and (a_1, e_1, v_1, s_1, c) have the same relation with $(\sigma_j, \rho_j, \delta_j, m_j, c)$ defined by the sign ing protocol. Therefore, given a signature $(\sigma_j, \rho_j, \delta_j, m_j, c)$, an infinitely powerful $\mathcal{S}^*$ can guess j correctly with probability exactly 1/2.

Theorem 3. *(Unforgeability) In the random oracle model, under the DLP assumption, the above partially blind signature is $(l, l+1)$-unforgeable against the sequential attack if $l < poly(\log \log q)$.*

Proof. The proof is similar to that in [2] except the knowledge extraction part. We first prove that after l interactions with the signer, the adversary user $\mathcal{U}^*$ can not forge a valid signature that the common information has never appeared in the interactions. This proof follows that used in [11]. By using $\mathcal{U}^*$, an interactive algorithm `Sim` is constructed to forge the plain version (without blindness) of the above scheme. Then, we use `Sim` to break the DLP (Discrete Logarithm Problem) assumption.

Let Q_F, Q_H denote the maximum number of queries asked from $\mathcal{U}^*$ to F and H, Q_S the maximum number of invocation of sign er $\mathcal{S}$. F and H reply with the

same values for the duplicated queries. Given the DLP-challenge (G, q, g, y), we are required to find $x = \log_g y \in \mathbb{Z}_q$. Sim simulates F, H, S as follows. Randomly select $I \leftarrow \{1, \cdots, Q_F + Q_S\}$ and $J \leftarrow \{1, \cdots, Q_H + Q_S\}$. For the i-th query, if $i = I$, return $z = F(c_I)$; else return $z = g^{\varpi_i}$, where $\varpi_i \in \mathbb{Z}_q$. For the j-th query to H, forward to H if $j = J$; else return a random integer in $\mathbb{Z}_q$. For query to $\mathcal{S}$, if common information $c \neq c_I$, simulate the signature by using witness ϖ_i, else the simulation fails. If $\mathcal{U}^*$ outputs valid signature pair $(\sigma, \rho, \delta, m, c)$, Sim outputs it. Since F and H may be asked during the simulation of the signer $\mathcal{S}$, it may have at most $Q_F + Q_S$ and $Q_H + Q_S$ points. The simulation of $\mathcal{S}$ is perfect if $c \neq c_I$ due to witness indistinguishability. Without any query to F, H, the adversary can successfully output a valid signature with negligible probability in $\log q$. Suppose that the success probability of $\mathcal{U}^*$ is at least $\epsilon_{\mathcal{U}^*}$, then the success probability of Sim is at least $\epsilon_{\mathtt{Sim}} = (Q_F + Q_S - 1)\epsilon_{\mathcal{U}^*}/(Q_F + Q_S)^2(Q_H + Q_S)^2$.

Now we construct an algorithm to solve $x = \log_g y \in \mathbb{Z}_q$ by using Sim as a subroutine. In this case, if Sim query to F, a random integer $\vartheta \in \mathbb{Z}_q$ is selected and yg^{ϑ} is returned to Sim. Note that query to F is before query to H. Now we use the standard rewind technique. After $1/\epsilon_{\mathtt{Sim}}$ trials, Sim will output a valid signature (σ, ρ, δ) with probability at least $1 - e^{-1}$ (here e is the base of natural logarithms). By repeating this rewind-trial $2/\epsilon_{\mathtt{Sim}}$ times, we obtain another valid signature $(\sigma', \rho', \delta')$, with probability at least $(1-e^{-1})/2$. Hence, with a constant probability and polynomial running time, we have two valid signatures with the same first transcript a. It follows that $\sigma + x\rho + (x+\vartheta)\delta = \sigma' + x\rho' + (x+\vartheta)\delta'$ and $\rho + \delta \neq \rho' + \delta'$. Hence, we extract that $x = (\sigma - \sigma' + \vartheta(\delta - \delta'))/(\rho + \delta - \rho' + \delta')$.

Then we consider an adversary $\mathcal{U}^*$ which outputs $l + 1$ valid signature with the same common information c. Similarly, we will construct an algorithm Sim to solve the DLP assumption by using $\mathcal{U}^*$ as a black box. Assume that a DLP-challenge (G, q, g, z) is given to Sim. First, Sim selects $b \in_U \{0, 1\}$ and sets $(y, z) = (g^x, zg^{\vartheta})$ if $b = 0$, or $(y, z) = (zg^{\vartheta}, g^w)$ if $b = 1$ by choosing ϑ and x (or w) randomly from $\mathbb{Z}_q$. F is defined so that it returns an appropriate value of z according to the choice. We assume that $b = 0$ is chosen and $(y, z) = (g^x, zg^{\vartheta})$ is set. Sim can then simulate signer $\mathcal{S}$, since the protocol between $\mathcal{S}$ and $\mathcal{U}^*$ is witness indistinguishable and $x = \log_g y$ is sufficient for Sim to complete the protocol. Let $\mathcal{S}'$ denote the signer simulated by Sim. If $\mathcal{U}^*$ is successful with probability at least $\epsilon_{\mathcal{U}^*}$, we can find a random tape string for $\mathcal{U}^*$ and $\mathcal{S}'$ with probability at least $1/2$ such that $\mathcal{S}'$ succeeds with probability at least $\epsilon_{\mathcal{U}^*}/2$. By employing $\mathcal{U}^*$ as a black-box, we can construct $\mathcal{U}'$, which has exactly the same interface with $\mathcal{S}'$ as $\mathcal{U}^*$ has, and plays the role of an impersonator in the interactive identification protocol with the verifier. When $\mathcal{U}^*$ asks at most q_F queries to random oracle H, $\mathcal{U}'$ is successful in completing the identification protocol with the verification procedure with probability at least $\epsilon_{\mathcal{U}^*}/2Q_H^{l+1}$, since, with probability greater than $1/2Q_H^{l+1}$, $\mathcal{U}'$ can guess a correct selection of $l + 1$ queries that $\mathcal{U}'$ eventually uses in the forgery. Sim then use the standard rewind technique for an interactive protocol to compute the discrete logarithm. Sim first runs $\mathcal{U}'$ with $\mathcal{S}'$ and the verifier, and find a successful challenge tuple $(\varepsilon_1, \cdots, \varepsilon_{l+1})$. Sim then randomly chooses an index, $i \in \{1, \cdots, l + 1\}$, and replays with the same environments and random tapes except different challenge

tuple $(\varepsilon_1, \cdots, \varepsilon_{i-1}, \varepsilon'_i, \ldots, \varepsilon'_{l+1})$ where the first $i-1$ challenges are unchanged. It follows that $\sigma_i + x\rho_i + (w_0+\vartheta)\delta = \sigma'_i + x\rho'_i + (w_0+\vartheta)\delta'_i$ and $\rho_i + \delta_i \neq \rho'_i + \delta'_i$. If $\delta_i \neq \delta'_i$, we can extract that $w_0 = (\sigma_i - \sigma'_i + x(\rho_i - \rho'_i))/(\delta'_i - \delta'_i) - \vartheta \bmod q$. Similarly, in the case of $b = 1$, if $\rho_i \neq \rho'_i$, we can extract $w_0 = (\sigma_i - \sigma'_i + w(\delta_i - \delta'_i))/(\rho'_i - \rho_i) - \vartheta \bmod q$. Let the event $\delta_i \neq \delta'_i$ happen with probability P. Since $\rho_i + \delta_i \neq \rho'_i + \delta'_i$, the event $\rho_i \neq \rho'_i$ happens with probability at least $1 - P$. Therefore, after finding a collision in the rewinding, we can extract w_0 with probability at least $0.5P + 0.5(1-P) = 1/2$ and solve the DLP-challenge. The remaining part to evaluate the collusion probability is the same as that in [2] and hence, it is omitted.

3.2 Partially Blind Inverse-Schnorr Signatures

As the above scheme, most DLP-based partially blind signatures exploit the witness indistinguishable signatures. In the following, we exploit the simplest solution to transform blind signatures to partially blind signatures by linking different public keys with different common information. However, we must address the issue to generate and manage the exponentially many public/private key pairs corresponding to the different common information. The trick is to use a publicly available deterministic algorithm to evolve the public key and enable the signer to evolve its private key accordingly. To illustrate this technique, we implement this transformation with the known blind Schnorr signature [22].

Let us assume that the same settings as the above scheme are used. Let G be a cyclic group with prime order q, and g a generator of G. $H, F : \{0,1\}^* \to \mathbb{Z}_q$ are public cryptographic hash functions. $y = g^x$ is the signer's public key and $x \in \mathbb{Z}_q$ is the corresponding secret key. Assume that the signer and the user agree on common information c. The resulting signature is a blind version of the knowledge signature $KS\{(x+F(c))^{-1} | g = (yg^{F(c)})^{(x+F(c))^{-1}}\}(m)$. The signature issuing protocol on the user's blind message m is as follows.

- (Key Evolution.) The signer computes $z = F(c)$, $Y = yg^z$ as its new public key and set $X = (x+z)^{-1} \bmod q$ accordingly as its new private key.
- (Initialization.) The signer randomly selects $r \leftarrow \mathbb{Z}_q$, and sends $a = Y^r$ to the user as a commitment.
- (Blinding.) The user computes $z = F(c)$, $Y = yg^z$. It picks random numbers $\alpha, \beta \leftarrow \mathbb{Z}_q$, computes $b = aY^\alpha g^\beta$, $\varepsilon = H(b||z||m)$, and returns a challenge $d = \varepsilon - \beta \bmod q$.
- (Signing.) The signer sends back $w = r - dX \bmod q$.
- (Unblinding.) The user computes $s = w + \alpha \bmod q$ and outputs (ε, s) as the resulting signature on m, and the pre-agreed common information c.
- (Verification.) The signature is valid if and only if $\varepsilon = H(Y^s g^\varepsilon || z || m)$ and $z = F(c), Y = yg^z$.

Clearly, except for an efficient additional key evolution procedure with c as the common information, the scheme is the same as the blind Schnorr signature. Hence, it enjoys the same security as the blind Schnorr signature. Since the blind Schnorr signature is not witness indistinguishable, its security is not proven with the standard DLP-assumption. Under the ROS assumption [15] (or see Appendix

Table 1. Comparison of the state-of-the-art partially blind signatures

	Signer	User	Verification	Length: bits	Security
PBS-AO	2 Exp	4 Exp	4 Exp	4λ	ROM+DLP
PBS-OR	2 Exp	3 Exp	3 Exp	3λ	ROM+DLP
PBS-Inv	1 Exp	2 Exp	2 Exp	2λ	ROM+GM+ROS

A), Schnorr proved that the scheme is secure in the random oracle model plus generic group model.

Theorem 4. *The above partially blind inverse-Schnorr Signature is correct and unconditionally partially blind, if the ROS assumption holds in the random oracle model plus the generic group model, the above partially blind signature is* $(l, l+1)$*-unforgeable.*

Proof. The correctness and the unconditional partial blindness are directly from those of the underlying blind Schnorr signature. After l interactions with the signer, the adversary $\mathcal{U}^*$ can not forge a valid signature that the common information has never appeared in the interactions. Or by using $\mathcal{U}^*$, another adversary $\mathcal{A}$ can be constructed to break the plain Schnorr signature (without blindness). The plain Schnorr signature has been proven secure in the random oracle model under the DLP assumption [13]. For an adversary $\mathcal{U}^*$ which outputs $l+1$ valid signature with the same common information c, the adversary $\mathcal{U}^*$ can be directly used to break $(l, l+1)$-unforgeability of the underlying blind Schnorr signature which has been proven secure if the ROS assumption holds in the random oracle model plus the generic group model [15] (The DLP assumption is weaker than the general group model assumption as it has been proven that DLP is difficult in the general group model [19].[20]). Hence, we obtain the above claim.

This transformation can be similarly extended to other DLP-related blind signatures such as the blind versions of Okamoto-Schnorr signature, ElGamal signature, DSA, etc. For instance, for the Okamoto-Schnorr signature with public key $y = g^{x_1}h^{x_2}$ and private keys $x_1, x_2 \in \mathbb{Z}_q$, the corresponding partially blind signature can be similarly built as the blind knowledge signature $KS\{\alpha, \beta | g = Y^{\alpha}h^{\beta}\}(m)$, where $Y = yg^z$, $\alpha = (x_1 + z)^{-1}$, $\beta = x_2(x_1 + z)^{-1}$, $z = F(c)$ and c is the pre-agreed common information. Clearly, after the key evolution, the partially blind signatures enjoy the same efficiency of the underlying blind signatures.

3.3 Comparison of Partially Blind Signatures

In this section, we give a comparison between the proposed two schemes (denoted by PBS-OR and PBS-Inv respectively) and the state-of-the-art partially blind signature [2] (denoted by PBS-AO) in terms of efficiency and security.

In the above table, Exp denotes the exponentiation in group G. $\lambda = \log q$. ROM stands for the random oracle model and GM means the generic group model. DLP represents the discrete logarithm problem in group G and ROS denotes the ROS problem introduced by Schnorr [15]. The computation complexity

is in terms of exponentiation without any optimization, that is, a two-base exponentiation is calculated as two single-base exponentiations and so on. Clearly, compared with the scheme in [2], our scheme PBS-OR is about 25% more efficient while enjoying the same level of security. The proposed PBS-Inv is the most efficient scheme among the three with a slightly stronger security assumption, i.e., the generic group assumption.

Note that the unforgeability of PBS-OR and PBS-AO is proven against the *sequential attack* formalized in [14]. The PBS-Inv is $(l, l+1)$-unforgeable against the *general parallel attack* formalized in [15]. However, it requires an additional ROS assumption. For (partially) blind signatures (including the PBS-AO scheme [2]) derived from knowledge proof of discrete logarithms, to our knowledge, *no* scheme has been proven secure against *generic parallel attack* (which is more powerful than the sequential attack) without relying on the ROS assumption which is a very strong assumption [16]. Although some scheme claimed such security [17], further analysis shows the claimed security is overestimated (Appendix A). On the other hand, these DLP-derived (partially) blind signatures can be proven secure against generic parallel attack under the ROS assumption using the similar techniques due to Schnorr [15]. From the state-of-the-art algorithm to solve the ROS problem [15], it requires subexponential running time $O(2^{2\sqrt{\log q}})$ [16] to break ROS problem which is less than previous evaluation $O(\sqrt{q})$ [15], where q is the order of group G. We must set a more conservative security parameter q satisfying $q > 2^{1600}$ rather than $q > 2^{160}$.

4 Conclusions

In this paper, efficient partially blind signatures are proposed with provable security. Compared with the scheme due to Abe *et al.*, our partially blind signatures based on witness indistinguishability enjoy the same level of security but are more efficient. We also suggest a simple yet efficient technique to transform blind signatures to partially blind signatures with the same security property as the underlying blind signatures. We implement this transformation with the known blind Schnorr signature.

Because of page limitation, we omit the additional security analysis. We refer the reader to the full version of this paper.

References

1. Abe, M., Fujisaki, E.: How to date blind signatures. In: Kim, K.-c., Matsumoto, T. (eds.) ASIACRYPT 1996. LNCS, vol. 1163, pp. 244–251. Springer, Heidelberg (1996)
2. Abe, M., Okamoto, T.: Provably secure partially blind signatures. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 271–286. Springer, Heidelberg (2000)
3. Cramer, R., Damgärd, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (1994)
4. Chaum, D.: Blind signatures for untraceable payments. In: McCurley, K.S., Ziegler, C.D. (eds.) Advances in Cryptology 1981 - 1997. LNCS, vol. 1440, Springer, Heidelberg (1999)

5. Chow, S.S.M., Hui, L.C.K., Yiu, S.M., Chow, K.P.: Two improved partially blind signature schemes from bilinear pairings. In: Boyd, C., González Nieto, J.M. (eds.) ACISP 2005. LNCS, vol. 3574, pp. 316–328. Springer, Heidelberg (2005)
6. Camenisch, J., Piveteau, J.-M., Stadler, M.: Blind signatures based on the discrete logarithm problem. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 428–432. Springer, Heidelberg (1995)
7. Fan, C.I., Lei, C.L.: Low-computation partially blind signatures for electronic cash. IEICE Transactions on Fundamentals of Electronics. Communications and Computer Sciences E81- A(5), 818–824 (1998)
8. Horster, H., Michels, M., Petersen, H.: Meta-message recovery and meta-blind signature schemes based on the discrete logarithm problem and their applications. In: Safavi-Naini, R., Pieprzyk, J.P. (eds.) ASIACRYPT 1994. LNCS, vol. 917, pp. 224–237. Springer, Heidelberg (1995)
9. Juels, A., Luby, M., Ostrovsky, R.: Security of blind digital signatures. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 150–164. Springer, Heidelberg (1997)
10. Okamoto, T., Ohta, K.: Divertible zero knowledge interactive proofs and commutative random self-reducibility. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 134–149. Springer, Heidelberg (1990)
11. Ohta, K., Okamoto, T.: On concrete security treatment of signatures derived from identification. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 354–369. Springer, Heidelberg (1998)
12. Pointcheval, D.: Strengthened security for blind signatures. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 391–405. Springer, Heidelberg (1998)
13. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology 13(3), 361–396 (2000)
14. Pointcheval, D., Stern, J.: Provably secure blind signature schemes. In: Kim, K.-c., Matsumoto, T. (eds.) ASIACRYPT 1996. LNCS, vol. 1163, pp. 252–265. Springer, Heidelberg (1996)
15. Schnorr, C.: Security of Blind Discrete Log Signatures against Interactive Attacks. In: Qing, S., Okamoto, T., Zhou, J. (eds.) ICICS 2001. LNCS, vol. 2229, pp. 1–12. Springer, Heidelberg (2001)
16. Wagner, D.: A generalized birthday problem. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 288–304. Springer, Heidelberg (2002)
17. Zhang, F., Kim, K.: Efficient ID-Based blind signature and proxy signature from bilinear pairings. In: Safavi-Naini, R., Seberry, J. (eds.) ACISP 2003. LNCS, vol. 2727, pp. 312–323. Springer, Heidelberg (2003)
18. Zhang, F., Safavi-Naini, R., Susilo, W.: Efficient verifiably encrypted signature and partially blind signature from bilinear pairings. In: Johansson, T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 191–204. Springer, Heidelberg (2003)
19. Nechaev, V.I.: Complexity of a determinate algorithm for the discrete logarithm. Mathematical Notes 55, 165–172 (1994)
20. Shoup, V.: Low bounds for discrete logarithems and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997)
21. Maitland, G., Boyd, C., Provably, A.: Secure Restrictive Partially Blind Signature Scheme. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 99–114. Springer, Heidelberg (2002)
22. Okamoto, T.: Provable secure and practice identifcation schemes and corresponding signature schemes. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 31–53. Springer, Heidelberg (1993)

Study on Grid-Based Special Remotely Sensed Data Processing Node

Jianqin Wang[1], Yong Xue[2,3,*], Yincui Hu[2], Chaolin Wu[2], Jianping Guo[2,4], Lei Zheng[2,4], Ying Luo[2,4], RuiZhi Sun[1], GuangLi Liu[1], and YunLing Liu[1]

[1] College of Information and Electric Engineering, China Agricultural University, P.O. Box 142, Beijing, 10083, China
[2] State Key Laboratory of Remote Sensing Science, Jointly Sponsored by the Institute of Remote Sensing Applications of Chinese Academy of Sciences and Beijing Normal University, Institute of Remote Sensing Applications, Chinese Academy of Sciences, P.O. Box 9718, Beijing 100101, China
[3] Department of Computing, London Metropolitan University, 166-220 Holloway Road, London N7 8DB, UK
[4] Graduate School of the Chinese Academy of Sciences, Beijing, China
Tian1.wang@163.com, y.xue@londonmet.ac.uk

Abstract. Everyday, space missions involve the download, from space to ground, of huge amount of raw images that are stored in the ground stations geographically distributed. It is a practical pressing task to process these data resource in real time or almost real time and to effectively share spatial information among remote sensing community. Grid technology can provide access to a global distributed computing environment via authentication, authorization, negotiation and security tools. This paper discusses the key technologies of Grid-based special remotely sensed data processing node. First, the concept of Grid-based special remotely sensed data processing node is introduced. Following is the architecture and functions of this node. Based on this architecture, the tasks scheduling algorithm is presented. Finally we introduce the computing resource meta-module information registry and renewal mechanism of the Grid-based special remotely sensed data processing node for Spatial Information Grid (SIG).

1 Introduction

Every day, dozens of satellites collect huge amount of images of the Earth, which are stored in ground stations geographically distributed. Remote Sensing data is characterized by largeness and instantaneousness. Naturally, it emerges three obvious problems for remote sensing community: (1) Remotely sensed data is very significant in some cases, such as fire and flood monitoring. So far real time processing in remote sensing confronts many difficulties in one single computer, or even impossibility. (2) Large amount of remotely sensed data processing module or algorithms, which are

* Corresponding author.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1106–1113, 2007.
© Springer-Verlag Berlin Heidelberg 2007

already exit, are the valuable resources in the Internet. They cannot be effectively used by the entire community but quietly stay in the disks of a few remote sensing specialists. Iterance of writing common remote sensing algorithms such as pre-processing algorithms "geometric correction" again and again will waste much time. (3) Large scale of remotely sensing sensed data geographically distributed need to be managed effectively.

Grid computing technology enables the virtualization of distributed computing and data resources such as processing, network bandwidth and storage capacity to create a single system image, granting users and applications seamless access to vast IT capabilities (http://cs.nju.edu.cn/people/yangxc/grid-computing.htm).

The Grid, itself, was born at a workshop at Argonne National Laboratory in September 1997, called "building a computational Grid", which was followed in 1998 by the publication of "The Grid: Blueprint for a New Computing Infrastructure" by Ian Foster of Argonne National Laboratory and Carl Kesselman of the University of Southern California (Foster and Kesselman 1998). "Grid" is used to refer to "a software system that provides uniform and location independent access to geographically and organizationally dispersed, heterogeneous resources that are persistent and supported" (Johnston et al. 2004). It has emerged as an important new field, distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and, in some cases, high-performance orientation (Foster et al. 2001). Foster et al. (2001) defined "Grid problem" as flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources –what they referred to as virtual organizations (VOs).

Today there are several Grid projects in remotely sensed information service. Aloisio *et al.* (2001) described how to provide comfortable, intuitive, yet powerful Web access to supercomputing. A Web-based, Grid-enabled application that processes, analyses, and delivers remote-sensing images provides an example of the technology at work. New image processing algorithms, which is distributed computing approach (Petrie *et al.* 2003), for remote sensed images are now being considered by Pacific Northwest National Laboratory. Aloisio *et al.* (2000) presented and discussed an architecture that allows transparent access to remote supercomputing facilities from a web gateway. The implementation exploits the Globus toolkit and provides users with fast, secure and reliable access to parallel applications. They showed the usefulness of their approach in the context of Digital Puglia, an active digital library of remote sensing digital data. Aloisio and Cafaro (2003) presented an overview of SARA/Digital Puglia (Synthetic Aperture Radar Atlas), a remote sensing environment that showed how Grid technologies and high performance computing can be efficiently used to build dynamic earth observation systems for the management of huge quantities of data coming from space missions and for their on-demand processing and delivering to final users. SARA/Digital Puglia is a Grid-enabled, high performance digital library of remote sensing images, developed in a joint research project with CACR/Caltech, ISI/USC and the Italian Space Agency. Grid-based Special Remotely Sensed Data Processing Node being built by Institute of Remote Sensing Applications, Chinese Academy of Science provides uniform super virtual computing power and share complex data processing module and algorithms in

order to let users conveniently get all the remotely sensed information service as well as do shopping.

In this paper, we will discuss the key technologies of Grid-based special remotely sensed data processing node. First, the concept of Grid-based special remotely sensed data processing node is introduced. Following is the architecture and functions of this node. Based on this architecture, the tasks scheduling algorithm is presented. Finally we introduce the computing resource meta-data information registry and renewal mechanism of the Grid-based special remotely sensed data processing node for Spatial Information Grid (SIG).

2 Architecture of Grid-Based Special Remotely Sensed Data Processing Node

Remotely sensed data is one of the important spatial information resources. So, the study on spatial information Grid must involve the content of Grid-based remotely sensed data processing node.

2.1 Conception of Grid-Based Special Remotely Sensed Data Processing Node

The conception of Grid-based special remotely sensed data processing node has two meanings from different point of views.

- Physical perspective:
  From the perspective of Grid topological structure, Grid-based special remotely sensed data processing node is the physical resource entity that constitutes the spatial information Grid environment. The hardware includes PCs and/or high performance computers and the network infrastructure. The software is composed of software resources for remotely sensed data processing and middleware resources.
- Logical perspective:
  The logical meaning of the node is similar to VOs. It can help to search and process remotely sensed data across locations and organizations. The whole process of the node involves the management and sharing of virtual resources across different locations and heterogeneous environments. With integration of the above virtual resources, the node has the ability of remotely sensed data processing. And what's more, it can provide an integrated virtual resource for spatial information Grid.

2.2 Architecture of Grid-Based Special Remotely Sensed Data Processing Node

As one of the nodes of Spatial Information Grid (SIG), the Grid-based remotely sensed data processing node can not only manage all the resources and tasks of SIG, but also provide the computing resource for spatial information Grid. To accomplish above purposes, we present the architecture of the Grid-based special remotely sensed data processing node. Figure1 shows the architecture of the node.

There are computing resource, resource monitor, global scheduler, job queue database, remote sensing algorithms and models database, RFT server, meta-modules

registering server and a serial of services in the architecture. The services include remote algorithm accepting service, resource monitoring service and so on. Meta-module service let users know about the knowledge of the remote sensing data processing algorithms of the node. More detail descriptions will be given in Section 4. In the Grid-based remotely sensed data processing node, RFT (Reliable File Transfer) service is responsible for the large amount of data transferring between the data server, node and users. The service can get data from remotely sensed data servers of spatial information Grid and provide the results for users. The node can provide processing algorithm resources. Also user can send their algorithms to the node to be executed when the algorithms provided by the node cannot accomplish user's purpose. Remote algorithm accepting service can help do this work. Resource monitoring service monitors all the computing resource of the node. The service collects status information of the computing resources and sends it to global task scheduler. The status information, which consists of the static and dynamic information, is useful for task scheduling. The global scheduler is responsible for manage all the tasks of the node. It is a vital component of the node. The tasks submitted to the node are immediately sent to job queue database. The tasks are recognized as records in database. When the task is finished, the record of the task will be deleted. There are two types of temporary files within the node, namely data temporary files and algorithm temporary files. The data received from data server of spatial information Grid is saved as data temporary files, which will be deleted after being processed. The codes accepted by remote algorithm accepting service is saved as temporary files, too. When users submit tasks, some other information, such as the estimated time and the path of data, are submitted at the same time. Tasks accepting service accepts the service request and the additional information. Figure 2 shows one example of tasks accepting service. The accepting service accepts the information and save it as the records of the tasks queue database.

3 Node Management and Tasks Scheduling

The resources of the node and submitted tasks should be managed well to maximize efficient of the node. Here we emphasize the tasks managing and scheduling.

The submitted tasks and additional information are accepted and saved in the tasks queue database by the tasks accepting service. There are two kinds of tables in the tasks queue database. The tasks are stored in the task description table and the additional information such as path of data is stored in the data description table. Each record in the data description table has a unique ID number, which is the same as the ID number of the corresponding tasks record in the task description table. The ID number is the primary key of the tables in the tasks queue database.

The task scheduler, which schedules the inner tasks of the node, accept the tasks, process the tasks, accept the data, send the results and manage the computing resource. Tasks scheduling is important for the node. The submitted tasks are stored in the tasks queue database. Then one of them will be picked out to execute based on some rules. The rules adopted by the node are described below. First the submitted

tasks are grouped by the remote sensing algorithm, which it uses. One group is set up for one algorithm. Then we pick up one task of each group to be executed. For each group, all the priority ratio of tasks in the group will be calculated. Then the tasks with the minimal priority ratio will be picked out. All the selected tasks will be executed in order.

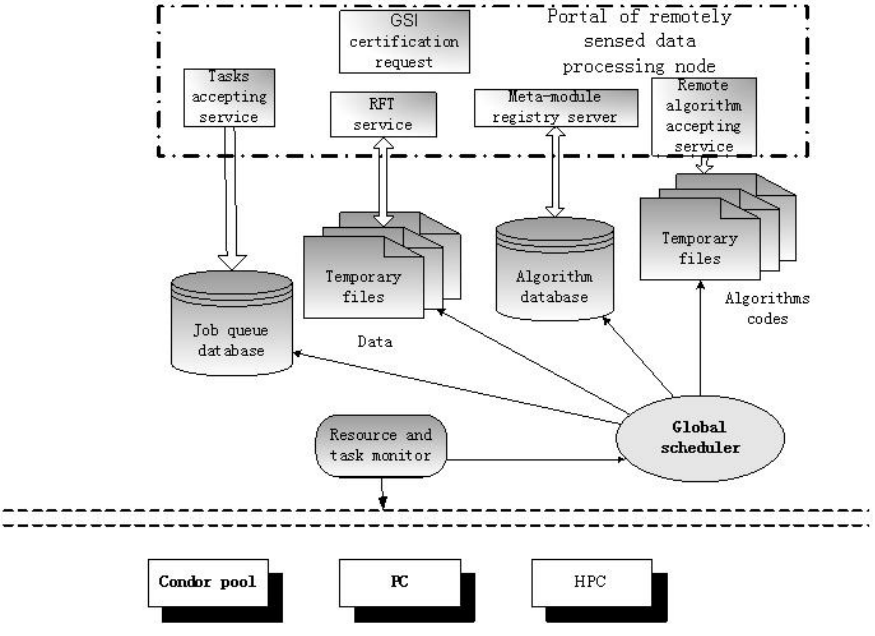


Fig. 1. Architecture of Grid-based Special Remotely Sensed Data Processing Node

4 Grid-Based Remote Sensing Meta-module Services

Grid technology is a very effective method for remotely sensed data processing service. Through it the existing remotely sensed data processing modules and algorithms resource can be shared seamlessly as a common service in Grid environment. It much more enhances the resource utility rate. In order to publish, describe, and manage Grid service data, the paper puts forward a Grid-based remotely sensed meta-module conception and corresponding structure. Combing the rule of this conception and OGSI technology, we define the Grid-based remotely sensed meta-module service data and Service Data Description language schema. Through remotely sensed Service Data Provider, the remote sensing service can be registered in the Registry Container of the special remotely sensed data processing Grid node. Users can also search and choose the proper services by demand line or Grid browser. After that, users can send a service request to Grid computing node according to the service data description.

```
ReceivingService  { Parameter
                    {Definition:
                        anyURL: InputDataPath;   float:Threshold value;
                        duration:Runtime;   String:ArithmeticName   ···
                      Handle
                    Definition:
                        ReceiveDataInfo(anyURL InputDataPath, String dataType, DataSize)
                            { ···      }
                        ReceiveThreshold(float Threshold value)
                            { ···      }
                        ReceiveRuntime(duration  Runtime)
                            { ···      }
                        ReceiveModelDescription(String ArithmeticName)
                            { ···      }
                        Other handle(otherparameterType otherparameter)
                          Local
                        Operation:
                          Oper1(Put the tasks into tasks queue database)
                          Oper2( Put the path of remotely sensed data into xml files )
                          Opern( Other operations )
                    } // ReceivingService
```

Fig. 2. Example of tasks accepting service

4.1 Concept of Grid-Based Remote Sensing Meta-module

The Service Data Element (SDE) model provides standard mechanisms for querying, updating and adding and removing data associated with each Grid service instance

We defined the concept of Grid-based remote sensing meta-module as follows: *The metadata that describe serials of remotely sensed data processing modules and algorithms in order to publish and manage these resources in Grid environment.* The description of Grid-based remote sensing meta-module must include function representations of algorithms, input and output descriptions, control parameter descriptions, etc. As Grid-based remote sensing meta-module, it must match these conditions:

- They must be described as the form of the Service Data Element (SDE) of Grid Service.
- Expression schema must match with the rule of the Grid Web Service Description Language (GWSDL)
- They can clearly describe the name, function, input and output parameter of the remotely sensed data processing module.
- The provided meta-module can be registered and updated in Grid environment.

Remotely Sensed meta-module SDD XML Schema is made up of six parts: domain name, version copyright, module information, module algorithm and running time, input information, output information.

4.2 Grid-Based Remotely Sensed Meta-module Service Registry

Combing Globus Open Grid Service Infrastructure (OGSI) rules and remotely sensed meta-module above, we provide the remotely sensed Grid service registry method.

Serials of remotely sensed data processing module service data, such as NDVI service, aerosol computing service, image classification service, etc., are registered in the registry container of the special remotely sensed data processing Grid node. For users, they can search the proper services from the service data through special portal made up of remote sensing service sets. There are two ways to find service through Grid service, one is based on demand line, and the other is Grid browser.

4.3 Grid-Based Remotely Sensed Meta-module Service Update

With the changes of remote sensing algorithms, the related meta-module information should be updated automatically. The registry container can accept the service update notifications and take actions including accepting the module update notification, deleting the old meta-module information and again registering of meta-module information.

The mechanism of update notification is push notification. Registry container subscribes to the notification of remote sensing meta-module update. If there is any information of module updating, the remote sensing SDE will send notification to the registry container.

Renewal information of meta-module will be registered at Registry Container in registry. Registry container can delete the old remotely sensed meta-module information and act as the observer.

Notification service instance of remotely sensed meta-module will be created by service factory. Registry container can get the notification information through describe service update information when the module renewing.

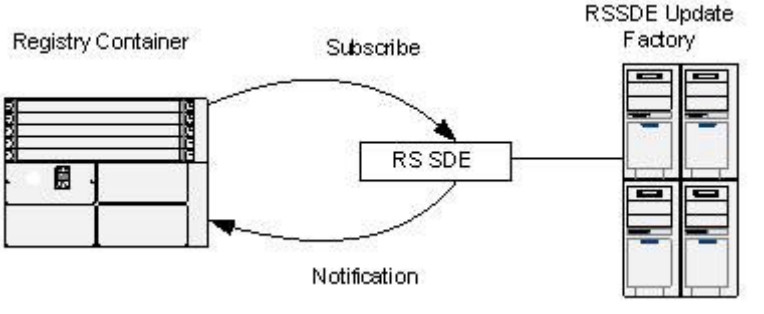


Fig. 3. Meta-module update based on Grid platform

5 Conclusions

Spatial Information Grid (SIG) is an infrastructure for spatial information collection and sharing by integrated management and processing in order to provide services as needed. SIG provides a basic technical architecture for integrated spatial data

acquisition, data processing and applications as well as an intelligent spatial data processing platform and basic environment. SIG aims to quickly process and analyse large amount of spatial data (from Terabyte to Petabyte) in near real time. Grid-based Special Remotely Sensed Data Processing Node is the important part of the SIG. The node can provide RFT service, tasks accepting service, meta-module registry service and renewal service. Task schedule mechanism provides effective scheduler algorithm. Remotely sensed meta-module register by Service Data of OGSA and update by Notification mechanism.

Acknowledgement

This publication is an output from the research projects "Grid platform based aerosol fast monitoring modeling using MODIS data and middlewares development" (40471091) funded by National Natural Science Foundation of China (NSFC), China, “Dynamic Monitoring of Beijing Olympic Environment Using Remote Sensing” (2002BA904B07-2) and “863 Program - Remote Sensing Information Processing and Service Node” (2003AA11135110) funded by the MOST, China, "Digital Earth" (KZCX2-312) funded by Chinese Academy of Sciences, China, and "Research Fund for Talent Program" funded by China Agricultural University.

References

Aloisio, G., Cafaro, M., Kesselman, C., Williams, R.: Web access to supercomputing. Computing In Science & Engineering 3(6), 66–72 (2001)

Aloisio, G., Cafaro, M., Falabella, P., Kesselman, C., Williams, R.: Grid computing on the web using the globus toolkit. In: Williams, R., Afsarmanesh, H., Bubak, M., Hertzberger, B. (eds.) High-Performance Computing and Networking. LNCS, vol. 1823, pp. 32–40. Springer, Heidelberg (2000)

Foster, I., Kesselman, C. (eds.): The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)

Foster, I.C.K., Tuecke, S.: The Anatomy of the Grid. Intl J. Supercomputer Applications (2001)

Johnston, W.E., Gannon, D., Nitzberg, B., Tanner, L.A., Thigpen, B., Woo, A.: Computing and Data Grids for Science and Engineering (2004), http://www.sc2000.org/techpapr/papers/pap.pap253.pdf

Petrie, G.M., Dippold, C., Fann, G., Jones, D., Jurrus, E., Moon, B., Perrine, K.: Distributed Computing Approach for Remote Sensing Data. In: Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS 2003) held in Nice, France, on 22-26 April 2003 (2003)

Xue, Y., Wang, J., et al.: Preliminary study of Grid computing for remotely sense information. International Journal of Remote Sensing 26(16), 3613–3630 (2005)

Novel Algorithms for Quantum Simulation of $3D$ Atom-Diatom Reactive Scattering

Ashot S. Gevorkyan[1], Gabriel G. Balint-Kurti[2], Alexander Bogdanov[3], and Gunnar Nyman[4]

[1] Institute for Informatics and Automation Problems, NAS of Armenia, P. Sevak St. 1, 375014 Yerevan, Armenia
g_ashot@sci.am
[2] University of Bristol, School of Chemistry, Bristol BS8 1TS, UK
Gabriel.Balint-Kurti@bristol.ac.uk
[3] Institute for High Performance Computing and Information Systems, 195251 St.-Petersburg, Russia
bogdanov@csa.ru
[4] Göteborg University, Department of Chemistry, SE-412 96, Göteborg, Sweden
nyman@chem.gu.se

Abstract. A new approach is described to the evaluation of the $\boldsymbol{S}$-matrix in three-dimensional atom-diatom reactive quantum scattering theory. A theory based on natural collision coordinates is developed, where the reaction coordinate can be viewed as fulfilling the same role as time in a time-dependent formulation. By writing the full wavefunction in coupled-channel form it is proved that the $3D$ multi-channel quantum reactive scattering problem can be treated in the same way as an inelastic single-arrangement problem. In particularly in the work two type coupled-channel representations, which lead to to two different systems of coupled-channel differential equations. The first system of coupled-channel equations is solved with the help of the $\boldsymbol{R}$-matrix propagation method yielding simultaneously the full wavefunction and all $\boldsymbol{S}$-matrix elements without further calculation. The second one is treated similarly. In this way we avoid a great volume of grid computations for $1D$ Schrödinger problem. The both algorithms use the intrinsic symmetry of scattering body-system, which allows to carry out maximally effective parallel computations of $3D$ scattering problem.

Keywords: quantum scattering, $\boldsymbol{S}$-matrix elements, coupled-channel differential equations, $\boldsymbol{R}$-matrix method, parallel algorithm.

1 Introduction

Accurate quantum dynamics approaches have made an immense impact on the theory of elementary atom-molecule collisions [1,2,3]. Recent advances in

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1114–1123, 2007.
© Springer-Verlag Berlin Heidelberg 2007

accurate quantum mechanical calculations on simple few atom systems have demonstrated the possibility of obtaining detailed information concerning chemical reactions from first principles [4,5,6]. Tremendous progress has occurred in developing and applying both time-independent and time-dependent quantum dynamical approaches in the last few years.

At present there are three common approaches used in the quantum reactive scattering field:

1. The hyperspherical coordinate methods [7,8],
2. The variational methods based on the simultaneous use of mass-scaled Jacobi coordinates in each of the chemical arrangements involved [9,10] and
3. Wavepacket methods [11,12] (many references can be found in the review [13]).

Despite the successful applications to polyatomic systems mentioned above, general applications still appear to be extremely difficult. Quantum mechanics is nonlocal in character and numerical efforts to solve the Schrödinger equation increase exponentially with the number of degrees of freedom. State resolved accurate quantum dynamical calculations are presently limited to seven-dimensional problems at most [14]. Thus, the development of new conceptual approaches giving rise to improved computational algorithms for direct quantum simulations is of great interest.

Earlier *natural collision coordinates* (NCC) allowing an analytical investigation of chemical reactivity in collinear collisions was introduced by Marcus [15] and further explored by Light [16,17]. Light and coworkers [18] generalized the NCC approach to three-dimensional atom-diatom reactive scattering and on this basis they studied H + H_2 system and its isotopomers in $3D$ for total angular momentum $J = 0$ [19].

Due to difficulties in applying the NCC approach, attention turned to other coordinate systems. Nevertheless, it seems to us that the primary idea of NCC, namely to simplify $3D$ quantum reactive scattering calculations, can be realized. Particularly, the authors has shown that for collinear three-body collisions, it is possible to construct the $\boldsymbol{S}$-matrix exactly, in a simple form [20,21].

The generalization of our previous collinear NCC quantum scattering approach to a full $3D$ treatment was made in the article [22].

In the present article two new algorithms for parallel modelling of atom-diatom reactions are proposed based on developed quantum scattering theory.

2 Schrödinger Equation for Reactive Scattering and Transition S-Matrix Elements

The reactive scattering process in a three-body system can be described by the scheme:

$$A + (BC)_n \longrightarrow \begin{cases} A + (BC)_m, \\ (AB)_m + C, \\ (AC)_p + B, \\ A + B + C, \\ \qquad\qquad (ABC)^* \longrightarrow \begin{cases} A + (BC)_m, \\ (AB)_m + C, \\ (AC)_p + B, \\ A + B + C, \end{cases} \end{cases} \qquad Sch.\ 1$$

where A, B, and C are atoms, n and m, p characterize quantum numbers of diatomic states corresponding to initial (in) and final (out) scattering channels, $(ABC)^*$ denotes an activated complex which may or may not be present as an intermediary species.

Quantum scattering in the three-body system after separation of external rotation in the body-fixed NCC system (intrinsic coordinates system (u, v, θ)) is described by the following equation for full wavefunction $\Phi^J_K(u, v, \theta)$:

$$\left\{ \left[\frac{\partial}{\partial u} \frac{1}{\eta^2} \frac{\partial}{\partial u} + \frac{\partial^2}{\partial v^2} \right] + \frac{q_0^2 + q_1^2}{q_0^2 q_1^2} \left[\frac{1}{\sin\theta} \frac{\partial}{\partial \theta} \sin\theta \frac{\partial}{\partial \theta} - \frac{K^2}{\sin^2\theta} \right] + \frac{2\mu}{\hbar^2} \big[E - U(u, v, \theta) \right.$$
$$\left. - E_{JK} \big] \right\} \Phi^J_K(u, v, \theta) + \Big\{ H^+_{JK} \Phi^J_{(K+1)}(u, v, \theta) + H^-_{JK} \Phi^J_{(K-1)}(u, v, \theta) \Big\} = 0, \quad (1)$$

where μ is a reduced mass of the three-body system, $q_0(u, v)$ and $q_1(u, v)$ are Jacobi coordinates, θ is the angle between vectors $\mathbf{q}_0$ and $\mathbf{q}_1$. The full interaction between the particles is described by the symbol $U(u, v, \theta)$. In Eq. (1) J is the total angular momentum, K is its projection in the body-fixed coordinate system, $\Phi^J_K(u, v, \theta)$ is the full wavefunction containing all projections of J (K describes projection in the (in) state). Moreover in the Eq. (1) the following notation are used:

$$E_{JK} = \frac{\hbar^2}{2\mu q_0^2} \Big(J(J+1) - 2K^2 \Big), \quad H^{\pm}_{JK} = \mp \frac{1}{q_0^2} C^{\pm}_{JK} \left\{ \frac{\partial}{\partial \theta} \pm (K \pm 1) \cot\theta \right\},$$
$$\eta(u, v) = \big(1 + K(u) v \big) \frac{ds}{du}, \qquad C^{\pm}_{JK} = \big[J(J+1) - K(K \pm 1) \big]^{1/2}. \quad (2)$$

In the Lamé coefficient $\eta(u, v)$ the symbol $K(u)$ is a curvature of the reaction coordinate and s is the length along it. It was shown, the transition $\boldsymbol{S}$-matrix in terms of the reaction coordinate (*intrinsic time*) may be calculated by the formula:

$$S^{J}_{\bar{n}\bar{j}\,\bar{K}\leftarrow njK}(E) = \lim_{u\to+\infty}\sqrt{\frac{k^{+}_{\bar{n}\bar{j}}}{k^{-}_{nj}}}\Big\langle \Phi^{(+)J}_{njK}(u,v,\theta)\Phi^{(out)J}_{\bar{n}\bar{j}\,\bar{K}}(u,v,\theta)\Big\rangle_{v\theta},$$

$$k^{\pm}_{\bar{n}\bar{j}} = \frac{1}{\hbar}\sqrt{2\mu(E-\epsilon^{\pm}_{\bar{n}(\bar{j})})}, \qquad \big\langle ...\big\rangle_{v\theta} = \int_{v}\eta^{1/2}\,dv\int_{\theta}\sin\theta\,d\theta, \tag{3}$$

where $\Phi^{(+)J}_{njK}(u,v,\theta)$ is the full wave function of three-body system and the solution of Eq. (1), which is evolved from the (in) state. $\epsilon^{+}_{n(j)}$ and $\epsilon^{-}_{n(j)}$ are the vibration-rotational energies in the (in) and (out) asymptotic channels respectively. The asymptotic wavefunction $\Phi^{(out)J}_{\bar{n}\bar{j}\,\bar{K}}(u,v,\theta)$ is described by:

$$\Phi^{(out)J}_{\bar{n}\bar{j}\bar{K}}(u,v,\theta) = \frac{1}{\sqrt{2\pi}}\exp\big(ik^{+}_{\bar{n}\bar{j}}\,u\big)\Upsilon^{(out)}_{\bar{n}(\bar{j})}(v)\Theta_{\bar{j}\,\bar{K}}(\theta). \tag{4}$$

In Eq. (4) the function $\Theta_{\bar{j}\,\bar{K}}(\theta)$ is a normalized associated Legendre polynomial, $\Upsilon^{(out)}_{\bar{n}(\bar{j})}(v)$ are the solutions of a $1D$ Schrödinger equation, which describes vibrational states taking rotation into account. It forms an orthonormal basis (see the solution of Eq. (7)).

The differential reactive state-to-state cross section may be simply constructed with help of $\boldsymbol{S}$-matrix elements by:

$$\sigma_{\bar{n}\bar{j}\,\leftarrow nj}(E,\vartheta) = \frac{1}{2j+1}\sum_{K\bar{K}}\frac{1}{4k^{2}_{nj}}\Big|\sum_{J}(2J+1)S^{J}_{\bar{n}\bar{j}\,\bar{K}\leftarrow njK}(E)d^{J}_{K\bar{K}}(\vartheta)\Big|^{2}, \tag{5}$$

where ϑ is the scattering angle and $k_{nj}\equiv k^{-}_{nj}$ is the incoming wave vector.

3 Coupled-Channel Method and Calculation of $\boldsymbol{S}$-Matrix Elements in Term of Intrinsic Time

Remembering that along u the system is in translational motion, while along v and θ the motion is localized, we employ the time-independent coupled-channel (CC) approach. It is convenient to write down the full wavefunction in the following form:

$$\Phi^{(+)J}_{K}(u,v,\theta) = \sum_{n'j'}G^{(+)J}_{n'j'K}(u)\Xi_{n'(j')}(v;u)\Theta_{j'K}(\theta). \tag{6}$$

The vibrational part of the wavefunction $\Xi_{n(j)}(v;u)$ forms an orthonormal basis in the variable v for each fixed value of u and satisfies the equation:

$$\left[-\frac{\hbar^{2}}{2\mu}\frac{d^{2}}{dv^{2}}+U(u,v)+\frac{\hbar^{2}j(j+1)}{2\mu v^{2}}\right]\Xi_{n(j)}(v;u) = \epsilon_{n(j)}(u)\,\Xi_{n(j)}(v;u), \tag{7}$$

where $U(u,v)$ is a some $2D$ effective potential, which in the limit of $u \to \mp\infty$ coincides with the asymptotic boundary states potential (see details in [22]). Note that Eq. (7) in the limit of $u \to \pm\infty$ transforms exactly into the equations, which describe (in) and (out) asymptotic channels. Taking into account (1) and (6) we can find the following second-order differential equation:

$$\begin{aligned}\Bigg\{\delta_{jj'}\delta_{KK'}\Bigg[\delta_{nn'}\frac{d^2}{du^2}+2\Big\langle\frac{\partial}{\partial u}-\frac{1}{\eta}\frac{\partial\eta}{\partial u}\Big\rangle_{nn'}\frac{d}{du}+\Big\langle\frac{\partial^2}{\partial u^2}-\frac{2}{\eta}\frac{\partial\eta}{\partial u}\frac{\partial}{\partial u}\Big\rangle_{nn'}\\ +\frac{2\mu}{\hbar^2}\Big\langle\eta^2\Big[E-E_{JK}-\epsilon_{n(j)}(u)+U(u,v)+\frac{\hbar^2 j(j+1)}{2\mu v^2}\Big]\Big\rangle_{nn'}\Bigg]\\ -\delta_{jj'}\Big\langle\frac{\eta^2}{q_0^2}\Big\rangle_{nn'}\Big[\delta_{K+1\,K'}\big(C^{+}_{JK'-1}\big)^2-\delta_{K-1\,K'}\big(C^{-}_{JK'+1}\big)^2\Big]\\ -\frac{2\mu}{\hbar^2}\Big\langle\eta^2 U^{K'}_{jj'}(u,v)\Big\rangle_{nn'}\delta_{KK'}\Bigg\}G^{(+)J}_{n'j'K'}(u)=0,\end{aligned}\tag{8}$$

where summation over the repeating index n is implied and we use the following notation for matrix elements:

$$\begin{aligned}\big\langle f(u)\big\rangle_{nn'}&=\int\Xi_{n(j)}(v;u)f(u,v)\Xi^{*}_{n'(j)}(v;u)dv,\\ U^{K}_{jj'}(u,v)&=\int\Theta_{jK}(\theta)\,U(u,v,\theta)\,\Theta_{j'K}(\theta)\sin\theta d\theta.\end{aligned}\tag{9}$$

Recall that in the (9) function $U(u,v,\theta)$ is the total interaction potential between all particles.

Thus we obtain a system of N coupled second-order differential equations. It is well known that this system can be rewritten in a more compact form as a system of $2N$ coupled first-order differential equations.

We now turn to the derivation of $\boldsymbol{S}$-matrix elements using the representation (6) for the full wavefunction. The full wavefunction of a three-body system is determined by a set of three quantum numbers (n,j,K). In order to describe the initial wavefunction with the help of representation (6) it is necessary to demand the asymptotic condition

$$\lim_{u\to-\infty}\sum_{n'j'}G^{(+)J}_{n'j'K}(u)\Xi_{n'(j')}(v;u)\Theta_{j'K'}(\theta)=\frac{1}{\sqrt{2\pi}}\exp\left(-ik_{nj}\,u\right)\delta_{nn'}\delta_{jj'},\tag{10}$$

to be fulfilled. This means that in the limit $u \to -\infty$, the wavefunction in Eq. (6) transforms to the asymptotical wavefunction $\Phi^{(in)J}_{njK}(u,v,\theta)$.

The $\boldsymbol{S}$-matrix elements with fixed quantum number J is obtained after substituting (6) into (3):

$$S^J_{\bar{n}\bar{j}\,\bar{K}\leftarrow njK}(E) = \sqrt{\frac{k^+_{\bar{n}\bar{j}}}{k^-_{nj}}} \lim_{u\to+\infty} \sum_{n'j'} \tilde{G}^{(+)J}_{n'j'K}(u)\, W_{n'\bar{n}}(u) \Lambda_{\bar{j}\,\bar{K}\leftarrow j'K}\,,$$

$$\Lambda_{\bar{j}\,\bar{K}\leftarrow j'K'} = \int_0^\pi \Theta_{\bar{j}\,\bar{K}}(\theta)\Theta_{j'K'}(\theta)\sin\theta\, d\theta,$$

$$W_{n'\bar{n}}(u) = \left\langle \Xi_{n'(j')}(v;u)\Upsilon^{(out)}_{\bar{n}(j')}(v)\right\rangle_v, \qquad \left\langle ...\right\rangle_v = \int_v \eta^{1/2}\, dv\, ..., \tag{11}$$

where the notation was maid $\tilde{G}^{(+)J}_{njK}(u) = \left(2\pi\right)^{-1/2} \exp(-ip^+_{nj}u) G^{(+)J}_{njK}(u)$ and it is simple to understand that in the limit of $u \to +\infty$ function $\tilde{G}^{(+)J}_{njK}(u)$ is constant.

The expressions (11) for $\boldsymbol{S}$-matrix elements can be simplified, if we take as a basis the functions $\Xi_{n(j)}(v;u)$, which in the limit $u \to +\infty$ coincide with the orthonormal basis wavefunctions $\Upsilon^{(out)}_{n(j)}(v)$. In this case we get the simplification $\lim_{u\to+\infty} W_{n'\bar{n}}(u) = \delta_{n'\bar{n}}$ and the following expression holds for $\boldsymbol{S}$- matrix elements:

$$S^J_{\bar{n}\bar{j}\,\bar{K}\leftarrow njK}(E) = \sum_{j'} \sqrt{\frac{k^+_{\bar{n}\bar{j}}}{k^-_{nj}}}\, \tilde{G}^{(+)J}_{\bar{n}j'\bar{K}}(+\infty) \Lambda_{\bar{j}\,\bar{K}\leftarrow j'K}\,. \tag{12}$$

The arbitrary $\boldsymbol{S}$-matrix elements of reactive scattering may be calculated simply with the help of formulas (3) and (6).

So, we have now shown that the initial quantum scattering problem can be exactly reduced to the standard set of N coupled second order differential equations (8) in a single variable u (where N is the number of channels or coupled equations). The matrix equation (8) has $2N$ linearly independent solutions. There are several standard ways in which this set of equations may be solved [3,16,23,24]. The $2N$ independent solutions are then combined to give N solutions which obey the boundary conditions as set out in Eqs. (10). The process of combining the independent solutions to satisfy the boundary conditions automatically yields the $\boldsymbol{S}$-matrix elements.

Computational Scheme A: The simulation algorithm on the grid consists of the following steps.

1. The parallel computation of one-dimensional Schrödinger equation (7) for the vibration states, taking account of the rotational motion on the grid (i.e. for a set of values u_i of the reaction coordinate).
2. The parallel computation of the matrix elements in Eq. (8) on the grid.
3. The solution of scattering equation (8) by $\boldsymbol{R}$-matrix method adapted for the NCC coordainate system and for the initial conditions, Eq. (10).
4. The calculation of the reactive scattering $\boldsymbol{S}$-matrix elements using Eq. (12) and the reaction cross sections, Eq. (5).

4 Etalon Equation Method, Full Wavefunction and *S*-Matrix Elements in Term of Intrinsic Time

For the development of the algorithm for numerical simulation, based on the aforementioned theoretical approach, is it necessary to solve equation (8) on the grid. The grid is formed along the reaction coordinate curve, which connects (*in*) and (*out*) scattering channels. The problem is that at each point of the grid one has to solve $M_i = n_i \times j_i$ number of one-dimensional quantum tasks and by further numerical integration to find the form of the matrix equation (8) on the grid. This procedure requires huge amount of computer resource and undergoes the accumulation of computation errors. Besides matrix equation (8) in this case will be given as a numerical array, which is extremely inconvenient for numerical simulation with changing integration step size. In order to overcome this difficulty one can use the method of exactly solvable etalon equation for the total wavefunction coupled-channel expression. This allows to get analytical form for one-dimensional matrix equation.

Another coupled-channel representation for full wavefunction can be written in the form:

$$\Phi_K^{(+)J}(u,v,\theta) = \sum_{\nu' j'} G^{(so)J}_{\nu' j' K}(u)\Xi^{(so)}_{\nu'(j')}(u,v)\Theta_{j' K}(\theta), \qquad 0 \le \nu' < \infty, \tag{13}$$

where $\Xi^{(so)}_{\nu(j)}(u,v)$ is the wavefunction of singular nonstationary (on intrinsic time u) quantum oscillator and satisfying the following equation (etalon equation):

$$i\hbar\frac{\partial \Xi^{(so)}(\tau,v)}{\partial\tau} = \left[-\frac{\hbar^2}{2\mu}\frac{d^2}{dv^2} + \left(\frac{\mu}{2}\omega^2(\tau)v^2 + \frac{g}{v^2}\right)\right]\Xi^{(so)}(\tau,v), \qquad 0 \le v < \infty,$$
$$\omega^2(\tau) = a_1 + a_2\tanh(\alpha\tau), \quad g = \frac{\hbar^2}{2\mu}j(j+1), \quad \tau = u/\omega_- a_0, \tag{14}$$

where a_0 is a Bohr radius, $a_1 > 0$, $a_2 > 0$ and α are some adjusting constants, which are chosen reasoning from calculation expediency, $\omega_- = \sqrt{a_1 + a_2}$.

The new etalon equation for vibration state (14) can be solved exactly [25]:

$$\Xi^{(so)}_{\nu(j)}(\tau,v) = \left[2\left(\frac{\mu}{\hbar\sigma^2}\right)^{a+1}\frac{\Gamma(n+1)}{\Gamma(n+a+1)}\right]^{1/2} v^{a+1/2} \times$$
$$\exp\left(-2in\gamma + \frac{i\mu}{2\hbar}\frac{\dot\sigma}{\sigma}v^2\right)L^a_\nu\left(\frac{\mu\dot\gamma}{\hbar}v^2\right), \tag{15}$$

were $\gamma(u)$ is the complex function and σ is a solution of classical oscillator problem:

$$\ddot\sigma + \omega^2(u)\sigma = 0, \qquad \dot\sigma = \frac{d\sigma}{d\tau}, \quad \dot\gamma = \frac{d\gamma}{d\tau}. \tag{16}$$

In the Eq. (15) $L_\nu^a(\hbar^{-1}\mu\dot{\gamma}v^2)$ is Laguerre polynomial and the wavefunctions $\Xi_{\nu(j)}^{(so)}(\tau, v)$ form an orthonormal basis: $\int_0^\infty \Xi_{\nu(j)}^{(so)}(uv_0^{-1}, v)\Xi_{\lambda(j)}^{(so)*}(uv_0^{-1}, v)dv = \delta_{\nu\lambda}$, where $v_0 = \omega_- a_0$.

Now comparing representations (6) and (13) is easy to find the following relation between $G_{njK}^{(+)J}(u)$ and $G_{\nu jK}^{(so)J}(u)$:

$$G_{njK}^{(+)J}(u) = \sum_{\nu'} C_{\nu n(j)}(u) G_{\nu' jK}^{(so)J}(u),$$

$$C_{\nu n(j)}(u) = \int_{-\infty}^{+\infty} \Xi_{\nu(j)}^{(so)}(uv_0^{-1}, v)\Xi_{n(j)}^{*}(v; u)dv. \tag{17}$$

Using (17), in the limit $u \to -\infty$ we can obtain the system of linear equations for definition of initial conditions of the expression (13):

$$\begin{cases} \sum_\nu C_{\nu 0(j)}(u) G_{\nu jK}^{(so)J}(u) = 0, \\ \sum_\nu C_{\nu 1(j)}(u) G_{\nu jK}^{(so)J}(u) = 0, \\ \qquad\qquad \cdots \qquad\qquad , \\ \sum_\nu C_{\nu n(j)}(u) G_{\nu jK}^{(so)J}(u) = o(u), \\ \sum_\nu C_{\nu n+1(j)}(u) G_{\nu jK}^{(so)J}(u) = 0, \\ \qquad\qquad \cdots \qquad\qquad , \\ \sum_\nu C_{\nu m(j)}(u) G_{\nu jK}^{(so)J}(u) = 0, \end{cases} \tag{18}$$

where $o(u) = (2\pi)^{-1/2}\exp(ik_{nj}^- u)$, n is a vibrational quantum number in the initial channel and m correspondingly is the number of vibrational states. Recall that in the limits $u \to \pm\infty$, the coefficient $C_{\nu n(j)}(u)$ is independent of the coordinate u.

In the system (18) the number of equations is finite while the number of unknown quantities is infinite. For exact definition of problem (18) we have to impose the following conditions:

$$G_{\nu jK}^{(so)J}(u) \equiv 0, \quad \nu > m. \tag{19}$$

Now taking into account (2) and (13) after simple analytical calculation we can get a new equation for reactive scattering:

$$\begin{aligned} \Bigg\{ \delta_{jj'}\delta_{KK'} \Bigg[\delta_{\nu\nu'}\frac{d^2}{du^2} + 2\left\langle \frac{\partial}{\partial u} - \frac{1}{\eta}\frac{\partial \eta}{\partial u} \right\rangle_{\nu\nu'} \frac{d}{du} + \left\langle \frac{\partial^2}{\partial u^2} - 2\left(\frac{1}{\eta}\frac{\partial \eta}{\partial u} + i\frac{\mu}{\hbar}v_-\right)\frac{\partial}{\partial u} \right\rangle_{\nu\nu'} \\ + \frac{2\mu}{\hbar^2}\left\langle \eta^2 \Big[E - E_{JK} + U(u,v) + \frac{\mu\omega^2}{2}v^2 + \frac{\hbar^2 j(j+1)}{2\mu v^2} \Big] \right\rangle_{\nu\nu'} \Bigg] \\ - \delta_{jj'}\left\langle \frac{\eta^2}{q_0^2} \right\rangle_{\nu\bar{\nu}} \Big[\delta_{K+1\,K'} \big(C_{J\,K'-1}^{+}\big)^2 - \delta_{K-1\,K'} \big(C_{J\,K'+1}^{-}\big)^2 \Big] \\ - \frac{2\mu}{\hbar^2}\left\langle \eta^2 U_{jj'}^{K'}(u,v) \right\rangle_{\nu\nu'} \delta_{KK'} \Bigg\} G_{\nu\, j'K'}^{(so)J}(u) = 0. \end{aligned} \tag{20}$$

Recall that in this case the summation over the repeating index ν is implied and we use the following notation for matrix elements:

$$\left\langle f(u)\right\rangle_{\nu\nu'} = \int \Xi^{(so)}_{n(j)}(uv_0^{-1}, v) f(u, v) \Xi^{(so)^*}_{\nu'(j)}(uv_0^{-1}, v) dv. \tag{21}$$

Finally substituting (13) in the (11) and taking into account (17) in the limit $u \to +\infty$ we find the new expression for $\boldsymbol{S}$-matrix elements:

$$S^J_{\bar{n}\bar{j}\,\bar{K} \leftarrow njK}(E) = \lim_{u\to+\infty} \sqrt{\frac{k^+_{\bar{n}\bar{j}}}{k^-_{nj}}} \sum_{\nu' n' j'} \Lambda_{\bar{j}\,\bar{K} \leftarrow j'K} C_{\nu' n'(j')}(u) W_{n'\bar{n}}(u) \tilde{G}^{(os)}_{\nu' j' K}(u)$$

$$= \sqrt{\frac{k^+_{\bar{n}\bar{j}}}{k^-_{nj}}} \sum_{\nu' j'} \Lambda_{\bar{j}\,\bar{K} \leftarrow j'K} C_{\nu'\bar{n}(j')}(+\infty) \tilde{G}^{(os)}_{\nu' j' K}(E, +\infty), \tag{22}$$

where $\tilde{G}^{(os)}_{\nu jK}(u) = \left(2\pi\right)^{-1/2} \exp(-ip^+_{nj}u) G^{(os)}_{\nu jK}(u)$ and in the limit of $u \to +\infty$ it is constant.

Computational Scheme B:

1. The parallel computation of the one-dimensional Schrödinger equation (7) for the vibrational states, taking into account the rotation in the (*in*) and (*out*) channels.
2. The solution of the system of linear equations (18).
3. The solution of scattering equation (20) by $\boldsymbol{R}$-matrix method and the computation of matrix elements $C_{\nu,n(j)}$ by means of the formula (17).
4. The calculation of transition $\boldsymbol{S}$-matrix elements using Eq. (22).
5. The computation of the reactive cross-section using formula (5).

5 Conclusion

New algorithms for the direct simulations of $3D$ atom-diatom reactive scattering have been developed within the framework of the natural collision coordinate (NCC) system. The NCC coordinate system allows us to reduce the initial $3D$ quantum reactive scattering problem to an inelastic single arrangement problem. In mathematical terminology, the problem is reduced to that of a matrix differential equation problem on a grid. The algorithms proposed (Schemas **A** and **B**) provide computational schemes which are maximally parallelized. However, the realization of the first step of the algorithm **A** is a very complicated computational problem. In order to implement the coupled differential equations (8) at every grid point, u_i, it is necessary to numerically solve $n \times j$ $1D$ Schrödinger problems. This problem is solved in the algorithm **B** using the etalon equation method (see (14)). However, in this case the initial conditions for scattering equation (20) are slightly complicated and are found only after the solution of the system of algebraic equations (18). Finally, it is significant that in comparison with other well-known computational algorithms both algorithms are favorable.

Acknowledgments

This work was partially supported by INTAS Grant No. 03-51-4000, and grant ICTS A-823.

References

1. Schatz, G.C., Kuppermann, A.: J. Chem. Phys., 65, 4642 (1976); ibid. 65, 4668 (1976)
2. Kuppermann, A.: In Theoretical Chemistry. In: Henderson, D. (ed.) Part A; Theory of Scattering: Papers in Honour of Henry Eyring, vol. 6, p. 79. Academic Press, New York (1981)
3. Manolopoulos, D.E., Clary, D.C.: Quantum calculations on reactive collisions. Annual Rep. C, The Royal Soci. of Chemistry 95 (1989)
4. Honvault, P., Launay, J.M.: J. Chem. Phys. 114, 1057 (2001)
5. Althorpe, S.C.: J. Chem. Phys. 114, 1601 (2001)
6. Gray, S.K., Balint-Kurti, G.G., Schatz, G.C., Lin, J.J., Liu, X., Harich, S., Yang, X.: J. Chem. Phys. 113, 7330 (2000)
7. Delves, L.M.: Nucl. Phys., 9, 391 (1959); 20, 275 (1960)
8. Schatz, G.C.: Chem. Phys. Lett. 150, 92 (1988)
9. Miller, W.H., der Jansen op, B.M.D.: A new basis set method of quantum scattering calculations. J. Chem. Phys., 86, 6213 (1987)
10. Baer, M., Kouri, D.J.: Phys. Rev. A., 4, 1924 (1971); J. Chem. Phys., 56, 4840 (1972); J. Math. Phys., 14, 1637 (1973)
11. Kosloff, R., Kosloff, D.: J. Phys. Chem. 79, 1823 (1983); Kosloff, R.: Time-dependent quantum-mechanical methods for molecular dynamics, J. Phys. Chem. 92, 2087 (1988)
12. Balint-Kurti, G.G.: Wavepacket theory of photodissociation and reactive scattering. Adv. Chem. Phys. 128, 244 (2003)
13. Nyman, G., Hua-Gen, Y.: Quantum theory of bimolecular chemical reactions. Rep. Prog. Phys. 63, 1001 (2000)
14. Yang, M., Zhang, D.H., Lee, S-Y.: J. Chem. Phys. 117, 9539 (2002)
15. Marcus, R.A.: J. Chem. Phys., 45, 4493 (1966); R. A. Marcus, J. Chem. Phys., 49, 2610 (1968)
16. Light, J.C., Walker, R.B.: J. Chem. Phys. 65, 4272 (1976)
17. Light, J.: Adv. Chem. Phys., 19, 1 (1971)
18. Frank, F.W., Light, J.C.: J. Chem. Phys., 90, 265 (1988)
19. Light, J.C., Altenberger-Siczek, A.: J. Chem. Phys., 88, 1824 (1988); J. Chem. Phys., 90, 265 (1988); J. Chem. Phys., 90, 300 (1988)
20. Bogdanov, A.V., Gevorkyan, A.S., Dubrovskiy, G.V.: Reduction of the collinear model of three-particle collisions with restructuring to the problem of an oscillator in an external force field. Tech. Phys. Lett., 20 N9, 698 (1994)
21. Gevorkyan, A.S.: Rep. NAS Armenia, 95 N3, 214 (1995)
22. Gevorkyan, A.S., Balint-Kurti, G.G., Nyman, G.: A new algorithm for $3D$ calculation of atom-diatom reactions (to be submitted)
23. Lester Jr., W.A.: In Modern Theoretical Chemistry. In: Miller, W.H. (ed.), vol. 1, Plenum Press, New York (1976)
24. Miller, W.H.: Annu. Rev. Phys. Chem., 41, 245 (1990)
25. Gamiz, P., et al.: J. Math. Phys., 10, 1422 (1969)
26. Skouteris, D., Castillo, J.F., Manolopoulos, D.E.: ABC: A Quantum Reactive Scattering Program. Comput. Phys. Commun. 133, 128 (2000)

An Algorithm for Rendering Generalized Depth of Field Effects Based on Simulated Heat Diffusion

Todd J. Kosloff[1] and Brian A. Barsky[2]

[1] University of California, Berkeley
Computer Science Division
Berkeley, CA 94720-1776
USA
koslofto@cs.berkeley.edu
[2] University of California, Berkeley
Computer Science Division and School of Optometry
Berkeley, CA 94720-1776
USA
barsky@cs.berkeley.edu

Abstract. *Depth of field* refers to the swath through a 3D scene that is imaged in acceptable focus through an optics system, such as a camera lens. Control over depth of field is an important artistic tool that can be used to emphasize the subject of a photograph. In a real camera, the control over depth of field is limited by the nature of the image formation process and by physical constraints. The depth of field effect has been simulated in computer graphics, but with the same limited control as found in real camera lenses. In this paper, we use *diffusion* in a non-homogeneous medium to generalize depth of field in computer graphics by enabling the user to independently specify the degree of blur at each point in three-dimensional space. Generalized depth of field provides a novel tool to emphasize an area of interest within a 3D scene, to pick objects out of a crowd, and to render a busy, complex picture more understandable by focusing only on relevant details that may be scattered throughout the scene. Our algorithm operates by blurring a sequence of nonplanar layers that form the scene. Choosing a suitable blur algorithm for the layers is critical; thus, we develop appropriate blur semantics such that the blur algorithm will properly generalize depth of field. We found that diffusion in a non-homogeneous medium is the process that best suits these semantics.

1 Introduction

Control over what is in focus and what is not in focus in an image is an important artistic tool. The range of depth in a 3D scene that is imaged in sufficient focus through an optics system, such as a camera lens, is called *depth of field* [1][2][3]. This forms a swath through a 3D scene that is bounded by two planes that are both parallel to the film/image plane of the camera, except in the case of the view camera [4][5][6][7][8][9].

In a real camera, depth of field is controlled by three quantities: the distance at which the camera lens is focused, the f/stop of the lens, and the focal length of the lens.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1124–1140, 2007.
© Springer-Verlag Berlin Heidelberg 2007

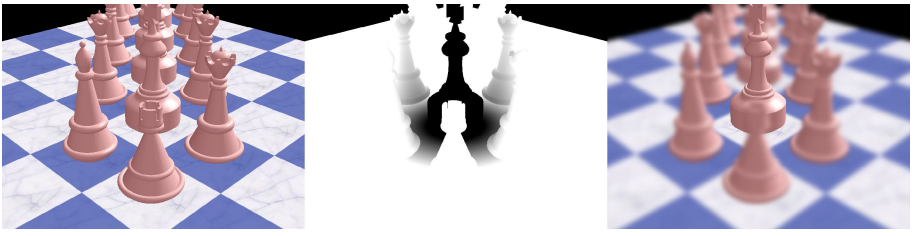


Fig. 1. Left: An unblurred chess scene rendered with a pinhole camera model. Middle: A cylindrical blur field. Right: Application of this blur field to the chess scene.

Professional photographers or cinematographers often control these adjustments to achieve desired effects in the image. For example, by restricting only part of a scene to be in focus, the viewer or the audience automatically attends primarily to that portion of the scene. Analogously, *pulling focus* in a movie directs the viewer to look at different places in the scene, following the point of focus as it moves continuously within the scene.

Creating computer generated imagery can be regarded as simulating the photographic process within a virtual environment. Rendering algorithms in computer graphics that lack depth of field are in fact modeling a pinhole camera model. Without depth of field, everything appears in completely sharp focus, leading to an unnatural, overly crisp appearance. Depth of field effects were first introduced into computer graphics as a means for increasing the realism of computer generated images. Just as in photography, depth of field is also used in computer graphics for controlling what portion of a scene is to be emphasized.

Real depth of field is discussed in Section 3.3. There are significant limitations on what can be achieved by adjusting the focus distance, f/stop, and focal length. It is evident that any given camera lens will have physical constraints that will limit the range of the adjustments. However, even if that were not the case, and we had at our disposal a magical lens that was unlimited in terms of these attributes, there are still many interesting and useful depth of field effects that could be imagined, but not realized by any combination of these adjustments, due to the nature of the image formation process.

When we allow our thinking to extend to what might be desirable, rather than what we have available, we can imagine useful depth of field effects that are not possible with conventional optics. Consider, for example, a scene consisting of a crowd of people. We want to draw attention to a person towards the front, and a second person towards the rear. We have to choose between focusing on the near person or the far person. If a sufficiently small aperture is available, we could perhaps get the entire crowd in focus, but this does not satisfy the goal of picking just the two people we have in mind. Instead, we would like to focus on the near person and the far person, while leaving the middle region out of focus. This cannot be done with a lens. Going even farther with this idea, consider a single row of people, arranged from left to right across the field of view. Since they are all at the same distance from the camera, they will all either be in focus, or all out of focus. There is no way to vary depth of field to focus in on any one of these people and not on the others.

In the case of computer graphics, we are still faced with precisely the same limitations as in real photography, since existing depth of field simulation algorithms were designed intentionally to precisely emulate real-world optics. We alleviate these limitations with the first implementation of an effect that we call *generalized depth of field*. In our system, focus is controlled not by the focus distance, f/stop, and focal length settings, but instead by a three-dimensional scalar blur field imposed on a scene. This approach enables the user to specify the amount of blur independently at every point in space.

Such a system overcomes many of the limitations on what can be achieved with focus. We can control the amount of depth of field independently from the amount of blur away from the in-focus region. Any number of different depths could be made to be in focus, while regions in between those depths will appear blurred. The amount of blur can be made to vary laterally, rather than only with depth. Even more exotic possibilities are just as easy with our system, such as contorting the in-focus region into the shape of a sphere, or any other geometric shape.

Our algorithm proceeds by first rendering the scene into a set of layers, which do not need to be planar. The 3D world coordinates of each pixel are stored in a position map associated with the layers. The position map is used to connect the user-specified 3D blur field to the layers. As is explained in Section 7.2 , blur values are also required that lie outside the object that is present within a layer. These blur values are extrapolated from the known values obtained through the aforementioned mapping. Having associated blur values with each pixel in the layers and having extrapolated those values, we use the blur values associated with each layer to blur the layers. Finally, the layers are composited from front to back, with alpha blending [10].

One difficulty is that naive spatially varying blur (a straightforward linear filter with a spatially varying kernel radius) produces artifacts in the form of holes appearing in the middle of objects. To solve this required a careful analysis of the true meaning of the blur field. This meaning, which we refer to as blur semantics, reflects our analysis of what we consider to be the essential characteristics of realistic depth of field that we wish to maintain, despite our goal of creating effects that cannot occur in the real world. The key insight is that the spatially variant blur operation that will satisfy our blur semantics is *diffusion in a non-homogeneous medium*.

2 Motivation for Generalized Depth of Field

2.1 Limitations of Real Depth of Field

When using depth of field as a tool for selecting only the relevant portions of a scene, there are limitations. If two people are next to each another, it is not possible to focus on just one or the other; both people are at the same distance from the camera, and hence both will be either in focus or out of focus. Alternatively, consider a crowd of many people. Perhaps we would like to use focus to highlight a few people scattered at a few locations throughout the crowd. There is clearly no combination of focus distance, f/stop, and focal length that can achieve this.

When simulating depth of field for computer generated images, the goal has generally been to faithfully replicate the behavior of real cameras, complete with these

limitations. On the other hand, we observe that there is no reason to accept these limitations in computer graphics. Rather than control the focus distance, f/stop, and focal length, we allow depth of field to be controlled by a 3D blur field that specifies how much blur should be applied to every point in 3D space.

2.2 Partial Occlusion

Since a lens has a finite aperture, that which is imaged at any given point on the film/image plane is the aggregate of the light emerging from every point on the lens. Thus, the lens can be considered as viewing the scene from many different points of view, simultaneously. Specifically, light rays impinging at the center of the lens may be emanating from foreground objects that occlude background objects whereas other rays arriving at a point on the lens far from its center may bypass the occluder. Thus, the background is visible from some points on the lens, but not from others. Therefore, there are single points on the image where a given object is *partially occluded*. This can be seen when objects that are out-of-focus have soft, semi-transparent edges. Partial occlusion is a vital aspect in the appearance of depth of field.

A computer simulation of depth of field that lacks partial occlusion does not faithfully reproduce the appearance of real photographs. Consider, for example, a small object that is close to the camera with a large aperture. Partial occlusion would be very noticeable because it would apply to the entire object and not just to the edge. In such cases, the lack of partial occlusion is completely unacceptable.

3 Background

3.1 Basic Lens Parameters

The focal length of a lens, denoted by f, is the distance the lens must be from the film/image plane to focus incoming parallel rays to a point on that plane. The aperture is the physical opening of the lens, which determines the quantity of light that will enter the optical system. We denote the aperture diameter by a_{diam}. The f/stop (or F number) is the ratio of the focal length to the diameter of the aperture of the lens, $\frac{f}{a_{diam}}$. Note that the aperture diameter is not completely specified by the f/stop. Indeed, when changing the focal length on a zoom lens, the f/stop varies in many amateur lenses, whereas it is the aperture that varies in in more professional zoom lenses.

3.2 Circle of Confusion

Blur arises when a point in the 3D scene is not imaged as a point, but instead spreads to form a disk, which is called the *circle of confusion*. The amount of blur for a given depth can be described by the diameter of the circle of confusion for that depth.

Consider a point that we wish to be in focus, located at a distance d_{focus} in front of the lens. Using the thin lens approximation [11], the distance d'_{focus} behind the lens where the film/image plane would have to be located can be derived [12]:

$$d'_{focus} = \frac{f * d_{focus}}{d_{focus} - f} \tag{1}$$

where f is focal length.

A point that is not at the distance d_{focus} in front of the lens will thus not be imaged as a point at the distance d'_{focus} behind the lens where the film/image plane is, but instead would form a point at the distance d_{image} behind the lens. Thus, on the film/image plane, it would be imaged as a circle of confusion having a diameter denoted by c_{diam}. Using similar triangles, this can be calculated [12]:

$$c_{diam} = a_{diam} * \frac{d_{image} - d'_{focus}}{d_{image}} \tag{2}$$

where a_{diam} is the aperture diameter. From this, we see that the circle of confusion diameter is intrinsically independent of focal length, although focal length would enter into the equation if it were recast in terms of f/stop rather than aperture diameter.

3.3 Real Depth of Field

In an optical system such as a camera lens, there is a plane in the 3D scene, located at the focus distance, that is rendered at optimal sharpness. There is a swath of volume through the scene that is rendered in reasonable sharpness, within a permissible circle of confusion, to be exact. This region of acceptable focus is delineated by near and far planes. However, this region is not centered at the focus distance; rather, the near plane is closer to the plane of perfect focus than is the far plane.

The particular distance that is imaged in perfect focus can be controlled by moving the lens towards or away from the film/image plane. Changing the focus distance will have a concomitant effect on the amount of depth of field, that is, the size of the swath. Specifically, for a given f/stop and focal length, focusing at a distance that is close to the camera provides only a narrow range of depths being in focus, and the amount of depth of field increases in a nonlinear fashion as the focus distance is increased, and conversely.

The amount of depth of field is also affected by The size of the aperture. The infinitesimal aperture of a pinhole camera has infinite depth of field, and this decreases as the aperture increases, and conversely.

Less widely discussed is the behavior of blur outside the region of acceptable focus. Objects that are at an increasing distance from the focus distance become increasingly blurred at a rate related to the aperture size. For very large aperture size, not only is there a very narrow range of depths that are imaged in focus, but the out-of-focus areas have an extreme amount of blur. This arises frequently in the case of macro-photography where the subject is very close to the camera. Although these relationships are well understood by experienced photographers, they are not obvious to novices. Furthermore, no amount of experience in photography can enable an escape from the limitations imposed by the inherent image formation process.

4 Related Work

The first simulation of depth of field in computer graphics was developed by Potmesil and Chakravarty who used a postprocessing approach where a single sharp image is

blurred using a spatially variant linear filter [13]. Postprocessing a single image can lead to occlusion artifacts. One such artifact occurs when background pixels are spread onto foreground pixels. Shinya solved this with a ray distribution buffer, or RDB. This approach, rather than blindly averaging adjacent pixels, stores pixels in a z-buffered RDB as they are being averaged. The z-buffer in the RDB ensures that near pixels will occlude far pixels during the averaging process. Rokita [14] achieved depth of field at rates suitable for virtual reality applications by repeated convolution with 3×3 filters and also provided a survey of depth of field techniques.

Cook introduced distributed raytracing [15], the first multisampling approach to depth of field. Several rays are traced for each pixel to be rendered. Each ray originates from a different point on the lens, and the rays are oriented such that they intersect at the plane of sharp focus. Kolb et al. performed distributed ray tracing through detailed lens models corresponding to actual lenses [16]. The resulting images exhibit the appropriate distortion and blur inherent in these lenses.

Haeberli and Akeley introduced the accumulation buffer [17], which is present on modern graphics hardware. The accumulation buffer allows several renders of a single scene to be accumulated. Depth of field is attained by accumulating aperture samples.

Scofield presents a fast and simple depth of field algorithm that, like ours, post-processes layers, then composites the layers from back to front [18]. His method, like ours, uses layers to solve the partial occlusion problem. Scofield's technique, unlike ours, assumes that each layer lies on a plane parallel to the film/image plane. Blur is uniform within a given layer so the blurring can be performed efficently using an FFT.

Kosara, Miksch, and Hauser were the first to suggest that by departing from the physical laws governing depth of field, blur can become a much more useful tool for indicating the relevant parts of a scene [19]. They call their system *semantic depth of field*, indicating that blur is a meaningful design element, rather than merely an artifact of the way lenses work. Their system operates in a similar manner to Scofield's. That is, objects are rendered into buffers, then the buffers are blurred according to the relevance of the objects, and finally the buffers are composited. This approach, while fast and simple, operates at object-level granularity. There is no way to blur only half an object, nor can blur vary smoothly across the surface of an object. Thus, it is impossible to have a window of focus smoothly move across a scene. On the other hand, we support a fully generalized blur field, where every point in 3D space can be blurred or not blurred as desired.

Depth of field is relevant in all optical systems, not only for camera lenses. The human eye also has a finite aperture (which is the pupil) and it can focus at different depths by means of accommodation of the internal crystalline lens. The optical characteristics of human eyes are slightly different for each person, resulting in different point spread functions from person to person. Barsky introduced vision realistic rendering to the computer graphics community [20], which uses actual data scanned from human eyes measured a wavefront aberrometer. The aberrometer samples a wavefront, which is then interpolated using the Zernike polynomial basis. The wavefront is converted into a set of point spread functions, which are then used to blur different parts of an image, using a postprocessing method. Further details about that postprocessing method are discussed in detail by Barsky et al. [21][22].

Isaksen et al. developed a new parameterization for light fields that enables controllable depth of field [23]. It is especially relevant here that their parameterization includes a focal surface that is not planar. Therefore, they can control focus in ways beyond what a real camera can produce. They do not completely generalize depth of field, however, focal surfaces are merely a subset of what can be achieved with our arbitrary blur fields.

Krivanek developed a very fast method for rendering point cloud models with depth of field via splatting [24]. This elegant method simply replaces each splat footprint with a Gaussian whose standard deviation increases with circle of confusion. Rendering these large splats does not lead to a slowdown, as it is possible to render large splats using a simplified version of the model comprising fewer points, without substantial loss in quality.

Perona and Malik used anisotropic diffusion to blur images in a nonuniform fashion [25][26]. Their purpose in doing so is for performing edge detection. We also use diffusion for blurring images in a nonuniform fashion, though for a completely different reason.

Bertalmio et al., [27] used diffusion to alleviate intensity leakage. They did not, however, address disocclusion of the background.

Whereas cameras whose film/image plane is parallel to the lens plane follow the conventional rules for predicting circle of confusion, view cameras do not follow these rules [5][6][7][8][9][28]. A view camera can be adjusted via tilts and swings to have its film/image and lens planes at arbitrary angles relative to one another. This results both in a change in the perspective projection and a change in depth of field. A view camera enables the plane of sharp focus to be oriented in a general manner such that it does not have to be parallel to the film/image plane. Thus, the plane that is in sharp focus might be an entire kitchen floor, even though the floor spans a great range of depths. Barsky and Pasztor developed methods for simulating the view camera camera model for computer generated images [4]. One approach used distributed ray tracing and the other used the accumulation buffer. Generalized depth of field can easily simulate a view camera, using an appropiate blur field as input.

For a more thorough survey of existing depth of field techniques, we refer the reader to a pair of surveys by Barsky et al. where the techniques have been separated into object space [12] and image space [29] techniques.

None of these depth of field algorithms provide a technique that can be modified to achieve our goal of arbitrary blur fields, as existing methods are too closely tied to the conventional image formation process to be coerced into producing the generalized blur effects we desire.

5 Blur Field Semantics

The selection of a 2D blur operation to apply to the layers is based not only on the need to apply arbitrary blur fields, but also on the need to smoothly generalize realistic depth of field.

The relevant property of depth of field relates to partial occlusion along the edges of an out-of-focus object. Since a camera lens has a finite aperture, the edges of a blurred

object become semi-transparent. We require this effect. However, we must decide how partial occlusion should behave in the presence of arbitrary blur fields. There is no physically-realizable camera or camera model to guide us here.

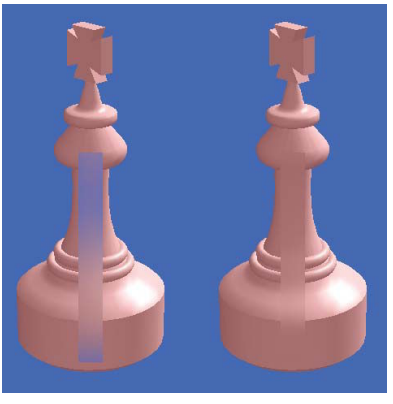


Fig. 2. Left: Naive linear filtering. Notice how the blurred region is transparent. Right: Diffusion. Notice how the blurred region remains opaque, as it should.

First, we explored a naive spatially variant linear filter on each layer. This results in images that are more blurred or less blurred in accordance with the blur field, and indeed we obtain partial transparency along the edges. However, transparency also occurred in isolated blurred regions interior to the foreground object. This does not correspond to anything present in realistic depth of field, and this is a highly objectionable artifact. Figure 2 (left) shows an example of such a hole.

We now examine the cause of this artifact and a solution to it. We represent transparency using RGBA pixel values [10], where transparent pixels have values of alpha less than 1. The initial unblurred image has some region in which the object lies. Inside this region, alpha is always 1 (in the case of opaque objects). Outside the object, the pixels have alpha of 0, indicating that nothing is there. During filtering, blurred pixels near the edge are averages of both the interior and the exterior of the object, leading to alpha values between 0 and 1. This results in a reasonable approximation to the partial occlusion found in depth of field. However, when the blur value inside an object is large, and the blur value towards the edges of the object are small, then the interior of the object may be averaged with pixels outside the object, despite the fact that the edge pixels themselves may remain perfectly sharp. We would avoid this artifact if the low-blur pixels on the edge acted as barriers through which colors cannot leak. This notion leads us to the key property that our 2D blur algorithm must possess: the averaging process must be aware of the blur values along the *entire path* that a color must traverse in order to reach a destination.

This property is inherently upheld by heat diffusion. Consider heat transfer on a sheet of metal. Making the analogy between heat and brightness in an image, imagine the idea of “painting a grayscale image” on the metal by heating some areas more than others. Over time, the heat diffuses, blurring the image. If the metal is more conductive

in some places and less conductive in others, then at any given time, some portions of the image will be more blurred than others. Most importantly, if there is a barrier of low conductivity between two regions of high conductivity, then no heat will flow from one region of high conductivity to the other. Therefore we take inspiration from heat diffusion in building our 2D blur operation. Figure 2 (right) shows how regions of low conductivity prevent a hole from appearing inside an object.

6 Algorithm Overview

In this section we describe the overall structure of our method. First, the scene is rendered as a collection of layers. Associated with each layer is a position map. This position map is required because the pixels within any given layer need not all occupy the same depth. Thus, our layers are more like 3D objects and less like the flat images that the term "layer" may suggest. A position map is similar to a depth map, except that it stores the full three-dimensional world coordinates of the object seen under each pixel. The position map can be visualized as a high dynamic range image, where red, green, and blue correspond to x, y, and z coordinates. Figure 3 shows a typical position map.

Fig. 3. A typical position map. This is a linearly scaled and clamped high dynamic range image. The red, green, and blue color channels represent x, y, and z world coordinates, respectively.

We will be applying a spatially varying blur operator to each layer, which is an inherently two-dimensional process. However, the blur field exists in three dimensions. We need a two-dimensional blur field for the layer blurring step. These two-dimensional blur fields are nonplanar slices of the three-dimensional blur field. The mapping between the two-dimensional blur field slice and the full three-dimensional blur field is performed by the aforementioned position map.

As will be explained in the next section, blur values are actually needed for some pixels outside the object itself. These pixels are those that lie within a certain distance of the object. We find these pixels by growing the region representing the object within

the layer. The position map inherently cannot contain any information for pixels outside the object; thus, these blur values cannot be read from the three-dimensional blur field. Therefore, reasonable blur values must be extrapolated from the known data.

Next, our carefully chosen, two-dimensional spatially varying blur operation is applied to each layer. Finally, the layers are composited from back to front, with alpha blending.

7 Algorithm Details

We now describe the region growing, blur field extrapolation, and diffusion steps in detail.

7.1 Region Growing

Often, only a small portion of a layer is covered. Thus, it would be a waste of computational resources to apply expensive blurring operations to the entire image when most of it is empty. On the other hand, a blurred object expands in size; thus, it is not sufficient only to blur the pixels lying inside the unblurred object. We therefore designate a superset of the object as the region to be blurred. This region is initialized to the object itself. The region is then grown one ring at a time by finding all the pixels lying outside the current region, but in contact with it. Rings are added until the region is sufficiently large to encompass any potential growth caused by the blurring operation. Figure 4 shows an object and the corresponding grown region. It is quite often the case that this expanded region is significantly smaller than the total size of the layer, resulting in a dramatic speedup compared to blurring the entire layer.

7.2 Blur Field Extrapolation

We now have blur values for each pixel within the object within the layer. However, our two-dimensional blur operation requires blur values outside the object, because a blurred image is larger than the original image. The region of the blurred image outside the original object is very important, since this is where the fuzzy edges characteristic of out-of-focus objects are. We cannot look up these blur values directly from the three-dimensional blur field, since there are no appropriate world coordinates present in the position map. Therefore, we must fabricate exterior blur values. This blur extrapolation step must ensure a smooth blur field that seamlessly merges with the interior of the object. Exterior blur values are calculated as a weighted average of the edge blur values, where the weights decrease with increasing distance from the edge point of interest. Note that the weights are normalized. That is, points farther from the edge are not less blurred; these points are merely influenced by the various edge points to a greater or lesser extent. Figure 5 illustrates what a typical extrapolated blur field looks like.

Consider the extrapolated blur value $B_{2D}(x,y)$ at a pixel (x,y) lying outside the object (but inside the expanded region). Let n be the number of pixels on the edge of the object, and g be a weighting function causing the relative contribution of each edge

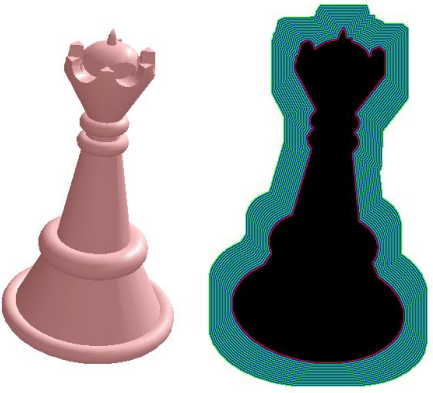


Fig. 4. A chess piece and the surrounding region within which blur will be applied

pixel to decrease with distance. Let (x_i, y_i) denote the i^{th} edge pixel. Then $B_{2D}(x,y)$ can be computed as:

$$B_{2D}(x,y) = \frac{\sum_{i=1}^{n} B_{2D}(x_i,y_i) * g(x_i,y_i,x,y)}{\sum_{i=1}^{n} g(x_i,y_i,x,y)}. \tag{3}$$

where $g = \frac{1}{d(x_i,y_i,x,y)^2}$, and $d(x_i,y_i,x,y)$ is the Euclidean distance between pixels (x_i,y_i) and (x, y); that is, $d = \sqrt{(x_i - x)^2 + (y_i - y)^2}$.

7.3 Blurring a 2D Layer

Two-dimensional spatially varying blur is performed by simulating heat diffusion over a medium with non-homogeneous thermal conductivity.

Whereas we might have started with the partial differential equation describing heat flow [30], then discretized it in space and numerically integrated it through time, our intent is to blur images, not to accurately predict the behavior of heat. Therefore, we use a simple implementation consisting of iterated neighborhood averaging, and we will not discuss the underlying differential equations any further.

A number of iterations are carried out. For each iteration, each pixel is averaged with its four neighbors, where the blur value for each pixel determines the weights in the average.

This neighborhood is iterated so that the object can be effectively blurred by kernels of any size. Repeated local averaging, rather than a single linear filter with a wider kernel, is necessary to prevent colors from leaking across regions of low blur.

Our blur operator works as follows: To calculate the blurred value of pixel (x,y) after one iteration, we average pixel (x,y) with its four neighbors, which we refer to as north, south, east, and west, and denote by C_n, C_s, C_e and C_w, respectively. The pixel at location (x,y) is referred to as "center", and is denoted by C_c. C_i denotes the image after i iterations. First, define W_n, W_s, W_e, and W_w to be the blur field elements, entries from B_{2D}, corresponding to the pixels north, south, east, and west of the pixel in question

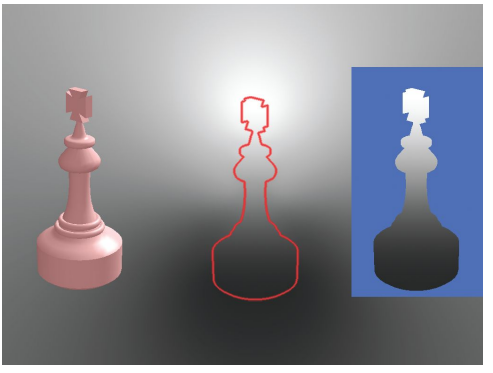


Fig. 5. Left: A layer. Right: The blur field slice for the layer. Middle: The blur values outside the object have been extrapolated.

Now we calculate C_{i+1}, the color of pixel (x,y) after the $(i+1)$'th iteration, using the following formulation:

$$C_c = C_i(x,y)$$

$$C_n = C_i(x,y-i)$$

$$C_s = C_i(x,y+1)$$

$$C_e = C_i(x+1,y)$$

$$C_w = C_i(x-1,y)$$

$$C_{i+1} = W_c * C_c + \tfrac{1}{4} * W_n * C_n + \tfrac{1}{4} * W_s * C_s + \tfrac{1}{4} * W_e * C_e + \tfrac{1}{4} * W_w * C_w$$

W_c is the weight for the center pixel. It is chosen such that all the weights will sum to 1.0, ensuring no unwanted brightening or dimming even when adjacent blur values are vastly different. That is, $W_c = 1.0 - \frac{W_n - W_s - W_e - W_w}{4}$.

8 Results: An Example

We have applied generalized depth of field to a chessboard scene. A fully in-focus version of this scene was rendered using a pinhole camera model. The scene was rendered into layers and corresponding position maps. The board itself is one layer, and each chess piece forms another layer. Figure 1 shows the original scene on the left, a blur field in the middle, and, on the right, the blurred scene corresponding to that blur field. In this figure, the blur field is a vertical cylinder of focus centered on one chess piece. Figure 6 shows a highly populated chessboard with a "plus"-shaped volume in focus. In Figure 7, we see a visualization of the chess scene intersecting with a blur field. This

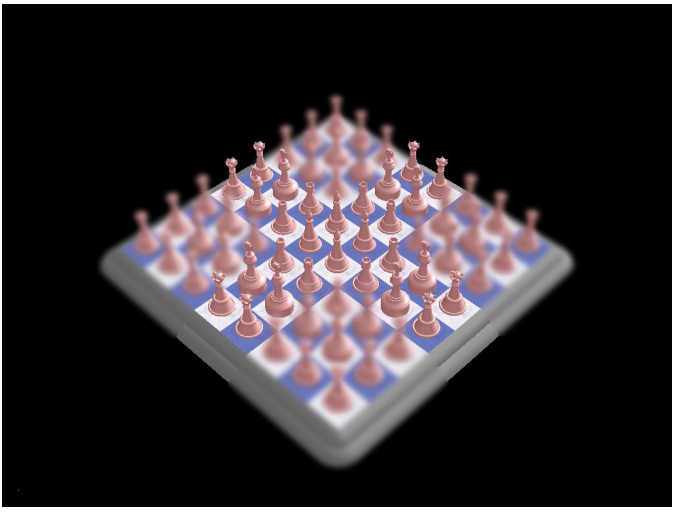


Fig. 6. The volume that is in focus is shaped like a "plus" sign

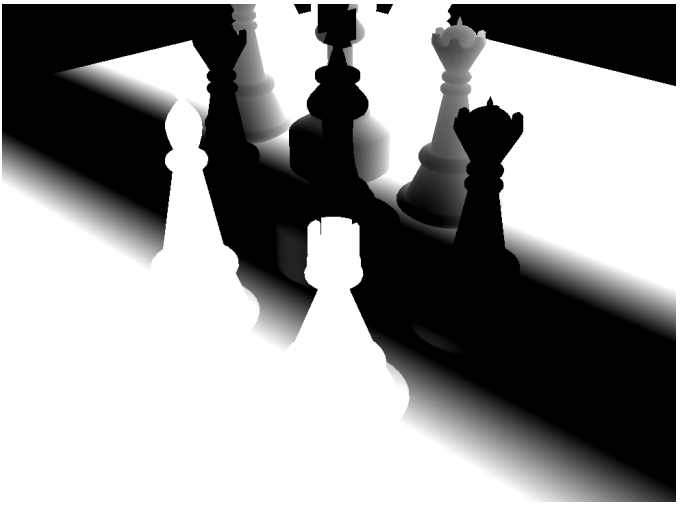


Fig. 7. This is a blur field that has a swath of acceptable focus which is planar, vertical, and aligned exactly with one row of the chessboard

blur field has a volume of focus oriented along a column of the chessboard. Figure 8 shows the blurred scene corresponding to the aforementioned blur field.

By varying the blur field over time, generalized depth of field can be used to create interesting variations on the traditional "pulling focus" For example, the cylindrical focal volume can smoothly glide from one chess piece to another. Alternatively, the

Fig. 8. The chess scene blurred with the swath of acceptable focus that was shown in Figure 7

scene can gradually come into focus as a sphere of focus emerges from the center of the scene and gradually expands in size.

9 Future Work

Currently, the scene must be split into a series of layers in order for our algorithm to achieve partial occlusion. Although any correct handling of partial occlusion would require some representation of depth complexity, our present approach does not work well for scenes that cannot easily be split into layers. Our layers are nonplanar, alleviating an important restriction in previous post-processing approaches to depth of field. The chessboard, for example, is one layer, while in previous methods it would have had to be split into many layers, because it spans a deep range of depths.

However, our method cannot presently handle a complex layer that occludes itself. Depth complexity is handled solely through different layers; hence, there cannot be any depth complexity within a single layer. This problem, however, is not unique to our formulation, but is an issue common to many post-processing approaches to depth of field.

A reasonable way to solve this would be to operate on a layered depth image (LDI) [31]. An LDI records the colors and depths of every object that is hit along each ray emanating from the center of projection. LDIs do not explicitly group the scene into objects or layers, but they do retain the colors of objects that are occluded. Thus, they have the benefits of layers without the disadvantages, and hence would make an excellent representation on which a future version of our algorithm could operate.

Generalized depth of field would be a useful tool in photography. We believe that by processing and combining a series of exposures taken with a camera over a variety

of focus settings, and with the help of computer vision techniques to estimate depth, generalized depth of field should be achievable for real world scenes.

10 Conclusion

In this paper, we have generalized depth of field in computer graphics by enabling the user to independently specify the amount of blur at each point in three-dimensional space. Generalized depth of field provides a novel tool to emphasize an area of interest within a 3D scene, to pick objects out of a crowd, and to render a busy, complex picture more understandable by focusing only on relevant details that may be scattered throughout the scene.

It is important to note that our method not only enables new effects, but it can also simulate conventional depth of field as a special case. This can be done easily, by constructing a blur field based on the circle of confusion of a conventional lens. A realistic blur field corresponding to a conventional camera would vary only along the depth axis, remaining constant within planes parallel to the image plane. The variation along the depth axis is related to the circle of confusion. Specifically, the blur value at a given depth is specified by equation 2 in section 3.2. Alternatively, a blur field corresponding to a view camera with tilted plane of sharp focus could be simulated by letting the circle of confusion vary along a tilted axis, rather than the depth axis.

Our approach has advantages over existing approaches for simulating realistic depth of field. Our approach is a post processing technique. Although other post processing methods (e.g. [13]) do not correctly handle occlusion, our approach does correctly handle occlusion, like the more expensive multisampling approaches. Unlike the other postprocessing methods, our approach is not based on the assumption that objects can be approximated as planar layers that are aligned parallel to the image plane, nor does our method suffer from discretization issues. Conversely, multisampling methods handle occlusion properly and do not suffer from discretization issues, but their performance degrades in proportion to the time it takes to render the scene. Thus, a scene that comprises many small polygons with complex shaders, which would already be slow to render due to its complexity, would need to be rendered many times in a multisampling method. On the other hand, postprocessing methods (such as ours) operate on an image based representation that is output by the renderer. Thus, our method requires no more time to process a scene of a million polygons than for a hundred polygons.

Acknowledgements

The chessboard model used to demonstrate our technique was created by Brian Sung Chul Choi and Jean Yi-Chin Lai.

References

1. Erickson, B., Romano, F.: Professional Digital Photography. Prentice-Hall, Englewood Cliffs (1999)
2. London, B., Upton, J., Kobre, K., Brill, B.: Photography, 7th edn. Prentice-Hall, Englewood Cliffs (2002)

3. Stroebel, L., Compton, J., Current, I.: Basic Photographic Materials and Processes, 2nd edn. Focal Press (2000)
4. Barsky, B.A., Pasztor, E.: Rendering skewed plane of sharp focus and associated depth of field. In: SIGGRAPH 2004 Tech Sketch, ACM Press, New York (2004)
5. Merklinger, H.M.: Focusing the View Camera (1993) ISBN 0-9695025-2-4
6. Merklinger, H.M.: View camera focus and depth of field, parts i and ii. View Camer magazine, July/August 1996, pp. 55-57 and September/October, pp. 56–58 (1996)
7. Simmons, S.: Using the View Camera, Amphoto Revised edn., Watson-Guptill Publications (1992)
8. Stone, J.: User's Guide to the View Camera, 3rd edn. Prentice-Hall, Englewood Cliffs (2004)
9. Stroebel, L.: View Camera Technique, 7th edn. Focal Press (1999)
10. Porter, T., Duff, T.: Compositing digital images. In: ACM SIGGRAPH 1984 Conference Proceedings, pp. 253–259 (1984)
11. Jenkins, F.A., White, H.E.: Fundamentals of Optics. McGraw-Hill, New York (1976)
12. Barsky, B.A., Horn, D.R., Klein, S.A., Pang, J.A., Yu, M.: Camera models and optical systems used in computer graphics: Part i, object based techniques. In: Kumar, V., Gavrilova, M., Tan, C.J.K., L'Ecuyer, P. (eds.) ICCSA 2003. LNCS, vol. 2667, pp. 246–255. Springer, Heidelberg (2003)
13. Potmesil, M., Chakravarty, I.: Synthetic image generation with a lens and aperture camera model. ACM Transactions on Graphics 1(2), 85–108 (1982)
14. Rokita, P.: Generating depth of-field effects in virtual reality applications. IEEE Computer Graphics and Applications 16(2), 18–21 (1996)
15. Cook, R.L., Porter, T., Carpenter, L.: Distributed ray tracing. In: ACM SIGGRAPH 1984 Conference Proceedings, pp. 137–145. ACM Press, New York (1984)
16. Kolb, C., Mitchell, D., Hanrahan, P.: A realistic camera model for computer graphics. In: ACM SIGGRAPH 1995 Conference Proceedings, pp. 317–324. ACM Press, New York (1995)
17. Haeberli, P., Akeley, K.: The accumulation buffer: hardware support for high-quality rendering. In: ACM SIGGRAPH 1990 Conference Proceedings, pp. 309–318. ACM Press, New York (1990)
18. Scofield, C.: $2\frac{1}{2} - d$ depth of field simulation for computer animation. In: Graphics Gems III, Morgan Kaufmann, San Francisco (1994)
19. Kosara, R., Miksch, S., Hauser, H.: Semantic depth of field. In: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01), p. 97 (2001)
20. Barsky, B.A.: Vision-realistic rendering: simulation of the scanned foveal image from wavefront data of human subjects. In: Proceedings of the 1st Symposium on Applied perception in graphics and visualization, pp. 73–81. ACM Press, New York (2004)
21. Barsky, B.A., Tobias, M.J., Horn, D.R., Chu, D.P.: Investigating occlusion and discretization problems in image space blurring techniques. In: First International Conference on Vision, Video, and Graphics, pp. 97–102 (2003)
22. Barsky, B.A., Tobias, M.J., Chu, D.P., Horn, D.R.: Elimination of artifacts due to occlusion and discretization problems in image space blurring techniques. Graphical Models 67(6), 584–599 (2005)
23. Isaksen, A., McMillan, L., Gortler, S.J.: Dynamically reparameterized light fields. In: ACM SIGGRAPH 2000 Conference Proceedings, ACM Press, New York (2000)
24. Krivanek, J., Zara, J., Bouatouch, K.: Fast depth of field rendering with surface splatting. In: Computer Graphics International 2003 (2003)
25. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. 12(7), 629–639 (1994)
26. Perona, P., Malik, J.: A network for multiscale image segmentation. In: IEEE Int. Symp. on Circuits and Systems, pp. 2565–2568. IEEE Computer Society Press, Los Alamitos (1988)

27. Bertalmio, M., Fort, P., Sanchez-Crespo, D.: Real-time, accurate depth of field using anisotropic diffusion and programmable graphics cards. In: IEEE Second International Symposium on 3DPVT, IEEE Computer Society Press, Los Alamitos (2004)
28. Shaman, H.: The view camera: operations and techniques. American Photographic Book Publishing Co. (1978)
29. Barsky, B.A., Horn, D.R., Klein, S.A., Pang, J.A., Yu, M.: Camera models and optical systems used in computer graphics: Part ii, image based techniques. In: Kumar, V., Gavrilova, M., Tan, C.J.K., L'Ecuyer, P. (eds.) ICCSA 2003. LNCS, vol. 2668, pp. 256–265. Springer, Heidelberg (2003)
30. Carslaw, H.S., Jaeger, J.C.: Conduction of Heat in Solids, 2nd edn. Oxford University Press, Oxford (1959)
31. Shade, J., Gortler, S.J., He, L.-w., Szeliski, R.: Layered depth images. In: Proceedings of ACM SIGGRAPH 1998, ACM Press, New York (1998)
32. Greenleaf, A.R.: Photographic Optics. The Macmillan Company, NYC (1950)
33. Ray, S.F.: Applied Photographic Optics, 3rd edn. Focal Press (2002)
34. Kingslake, R.: Optics in Photography. SPIE – The International Society for Optical Engineering (1992)
35. Williams, J.B.: Image Clarity: High-Resolution Photography. Focal Press (1990)
36. Goldberg, N.: Camera Technology: The Dark Side of the Lens. Academic Press, London (1992)
37. Shinya, M.: Post-filtering for depth of field simulation with ray distribution buffer. In: Proceedings of Graphics Interface '94, Canadian Information Processing Society, pp. 59–66 (1994)
38. Demers, J.: Depth of field: A survey of techniques. In: GPU Gems, pp. 375–390 (2004)
39. Merklinger, H.M.: The Ins and Outs of Focus: Internet Edition (1992), `http://www.trehnholm.org/hmmerk/download.html`
40. Barsky, B.A., Bargteil, A.W., Garcia, D.D., Klein, S.A.: Introducing vision-realistic rendering. In: Eurographics Rendering Workshop, posters, pp. 26–28 (2002)
41. Baker, A.A.: Applied Depth of Field. Focal Press (1985)

Fingerprint Template Protection Using Fuzzy Vault

Daesung Moon[1], Sungju Lee[2], Seunghwan Jung[2], Yongwha Chung[2,*], Miae Park[3], and Okyeon Yi[3]

[1] Biometrics Technology Research Team, ETRI, Daejeon, Korea
daesung@etri.re.kr
[2] Department of Computer and Information Science, Korea University, Korea
{peacfeel,sksghksl,ychungy}@korea.ac.kr
[3] Department of Mathematics, Kookmin University, Korea
{miae34,oyyi}@kookmin.ac.kr

Abstract. Biometric-based authentication can provide strong security guarantee about the identity of users. However, security of biometric data is particularly important as the compromise of the data will be permanent. To protect biometric data, we need to store it in a non–invertible transformed version. Thus, even if the transformed version is compromised, the actual biometric data remains safe. In this paper, we propose an approach to protect fingerprint templates by using the idea of the *fuzzy vault*. Fuzzy vault is a recently developed cryptographic construct to secure critical data with the fingerprint data in a way that only the authorized user can access the secret by providing the valid fingerprint. We modify the fuzzy vault to protect fingerprint templates and to perform fingerprint verification with the protected template at the same time. This is challenging because the fingerprint verification is performed in the domain of the protected form. Based on the experimental results, we confirm that the proposed approach can perform the fingerprint verification with the protected template.

Keywords: Crypto-Biometric, Template Protection, Fuzzy Vault.

1 Introduction

The increasing demand for more reliable and convenient security systems generates a renewed interest in human identification based on biometric identifiers such as fingerprints, iris, voice, signature and gait[1-2]. Since biometrics cannot be lost or forgotten like passwords, biometrics has the potential to offer higher security and more convenience for the users.

A common approach to biometric authentication is to capture the biometric measurements of users, transform the measurements into templates, and store the templates in a reference database during the *enrollment phase*. During the *verification phase*, a new measurement is captured and matched against the database information. Note

* Corresponding author.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1141–1151, 2007.
© Springer-Verlag Berlin Heidelberg 2007

that, biometric measurements are rarely identical and each measurement is environment- and device-dependent. Also, a template provides an approximate version of the biometric data, and verification succeeds if the second measurement is close to the stored template. And, in this paper, the fingerprint has been chosen as the biometrics for user authentication because it is more mature in terms of the algorithm availability and feasibility[1].

The fact that biometric templates are stored in a database introduces a number of security risks, and the following threats can be identified[1-6]:

1. *Impersonation*. An attacker steals templates from a database and constructs artificial biometrics that pass authentication.
2. *Irrevocability*. Once compromised, biometrics cannot be updated, reissued or destroyed.

For example, when an authentication system is used on a large-scale, the reference database has to be made available to many different verifiers, who, in general, cannot be trusted. Especially, in a networked environment, attacks on the database pose a serious threat. It was shown explicitly by Matsumoto et al.[7] that using information stolen from a database, artificial biometrics could be constructed to impersonate people. Construction of artificial biometrics is possible even if only part of the template is available. Hill[8] showed that if only minutiae templates of a fingerprint are available, it is still possible to successfully construct artificial biometrics that pass authentication.

The second threat was first addressed by Schneier[9]. Even without database compromise, biometric possession by government provides the potential for information misuse as the data belonging to the same user can be easily linked. In order to protect against the threats given above, we need to store a transformed version of the biometric data. The transformation is one-way and so knowledge of a transformed biometric does not leak information about the actual biometric data.

In this paper, we propose an approach to protect fingerprint templates by using the idea of the fuzzy vault[10]. *Fuzzy vault*, as explained in later sections, is a recently developed cryptographic construct, and has the characteristics that make it suitable for applications combining biometric authentication and cryptography: the advantages of cryptography(*e.g.,* proven security) and fingerprint-based authentication(*e.g.,* user convenience, non-repudiation) can be utilized in such systems.

We modify the fuzzy vault to protect fingerprint templates and to perform fingerprint verification with the protected template at the same time. This is challenging because the fingerprint verification is performed in the domain of the protected form. Based on the idea of the fuzzy vault, we first generate a minutiae table, which includes the non–invertible transformed version of the enrolled fingerprint data, and then perform fingerprint verification on the minutiae table with the input fingerprint data. Based on the experimental results, we confirm that the proposed approach can perform the fingerprint verification with the protected template securely, and cryptography and biometrics can be merged to achieve high security and user convenience at the same time.

The rest of the paper is structured as follows. Section 2 explains previous results to protect biometric data and the overview of the fuzzy fingerprint vault, and Section 3 describes the proposed approach to perform fingerprint verification with the protected template. The experimental results are given in Section 4, and conclusions are made in Section 5.

2 Background

2.1 Fingerprint Verification

A fingerprint authentication system has two phases: *enrollment* and *verification*[11]. In the off-line enrollment phase, an enrolled fingerprint image is preprocessed, and features, called as *minutiae*, are extracted and stored. In the on-line verification phase, the similarity between the enrolled template minutiae and the input minutiae is examined.

`Pre-Processing` refers to the refinement of the fingerprint image against the image distortion obtained from a fingerprint sensor. It consists of three stages. Binary conversion stage applies a low-pass filter to smooth the high frequency regions of the image and threshold to each sub-segment of the image. Thinning stage generates a one-pixel-width skeleton image by considering each pixel with its neighbors. In positioning stage, the skeleton obtained is transformed and/or rotated such that valid minutiae information can be extracted.

`Extraction` refers to the extraction of minutiae in the fingerprint image. After this step, some of the minutiae are detected and stored into a template file. A minutia can be specified by its coordinates, angle, and its type(ending/bifurcation).

Based on the minutiae, the input fingerprint is compared with the template fingerprint. Actually, `Match` is composed of alignment stage and matching stage. In order to match two fingerprints captured with unknown direction and position, the differences of direction and position between two fingerprints are detected, and alignment between them needs to be accomplished. Therefore, in alignment stage, transformations such as translation and rotation between two fingerprints are estimated, and two minutiae are aligned according to the estimated parameters. If alignment is performed accurately, matching stage is referred to point matching simply. That is, two minutiae are compared based on their position, orientation, and type. Then, a matching score is computed.

As explained in Section 1, some problems with fingerprint verification systems can arise when the template minutiae has been compromised. For verification systems based on physical tokens such as keys and badges, a compromised token can be easily canceled and the user can be assigned a new token. Similarly, user IDs and passwords can be changed as often as required. Yet, the user only has ten fingers. Thus, if the minutiae are compromised, the user may quickly run out of the minutiae to be used for authentication and cannot re-enroll forever. To solve this problem, some results have been reported.

2.2 Protection of Biometric Information

Ratha, Connell, and Bolle[12] introduced the term “cancelable biometrics” to protect biometric templates. Designing cancelable biometrics had many objectives. First, a cancelable template stored in a database of certain application cannot be used as a template in another application. Second, if a database record(a fingerprint template) is compromised, a new database record can be issued(just like a new password can be issued). Finally, altering a database record(replacing a fingerprint template) is unfeasible because the template can be digitally signed by the issuer, or some privileged

information(*e.g.,* an encryption key) can be stored in the template in such a way that it can be released only through biometric recognition. Although they explained their idea with a high-order polynomial function, it was conceptual and they did not implement their idea.

There were other innovative, yet similar methods that did not perform biometric matching. The first is the *fuzzy commitment* scheme[13]. Here, a secret(presumably a private key used for later authentication) is encoded using a standard error correcting code such as Hamming or Reed-Solomon, and then XOR-ed it with a biometric template. To retrieve the secret, a slightly different biometric template can again be XOR-ed, and the result put through an error correcting decoder. Some small number of bit errors introduced in the key can be corrected through the decoding process. The major flaw of this system is that biometric data is often subject to reordering and erasures, which cannot be handled by using this simple scheme.

A second paper[14] based on previous work on a fuzzy commitment scheme had a similar theoretical foundation to this work, but aimed toward a completely different application. Dodis, et al., demonstrated how such data could be used to generate strong keys for any kind of cryptographic application. They used the notion of a fuzzy extractor to describe the process of extracting a random string $\boldsymbol{U}$ from a biometric input b, in such a way that a certain amount of error was allowed for. If the input changes slightly to b, then the extracted $\boldsymbol{U}$ will be the same. To enable the recovery of U from b, the fuzzy extractor also outputs a public string $\boldsymbol{V}$. Additionally, Dodis described three metrics to measure the variation in the biometric reading: Hamming Distance, Set Distance and Edit Distance.

Recently, Juels and Sudan[16] proposed the *fuzzy vault*, a new architecture with applications similar to Juels and Wattenberg's fuzzy commitment scheme, but is more compatible with partial and reordered data. In the fuzzy commitment, Alice can place a secret value k(*e.g.,* private encryption key) in a vault and lock(secure) it using an unordered set A. Bob, using an unordered set B, can unlock the vault (access k) only if B overlaps with A to a great extent. The procedure for constructing the fuzzy vault is as follows: First, Alice selects a polynomial p of variable x that encodes k(*e.g.,* by fixing the coefficients of p according to k). She computes the polynomial projections, $p(A)$, for the elements of A. She adds some randomly generated "chaff" points that do not lie on p, to arrive at the final point set R. When Bob tries to learn k (*i.e.,* finding p), he uses his own unordered set B. If B overlaps with A substantially, he will be able to locate many points in R that lie on p. Using error-correction coding, it is assumed that he can reconstruct p(and hence k). The security of the scheme is based on the infeasibility of the polynomial reconstruction problem(*i.e.,* if Bob does not know many points that lie on p, he can not feasibly find the parameters of p, hence he cannot access k). Note that, since this fuzzy vault can work with unordered sets(common in biometric templates, including fingerprint minutiae data), it is a promising candidate for crypto-biometric systems.

Based on the fuzzy vault, some implementations results for fingerprint have been reported. For example, Clancy, et al.[15] and Uludag, et al.[16] proposed a *fuzzy fingerprint vault*. Similarly, Barral, et al[17] added the chaff points to the user's fingerprint information. Note that, their systems inherently assume that fingerprints(the one that locks the vault and the one that tries to unlock it) are pre-aligned. This is not a realistic assumption for fingerprint-based authentication schemes, and limits the

applicability of their schemes[16]. For the purpose of explanation, the fuzzy fingerprint vault is shown in Fig. 1, and the details of it can be found in [16].

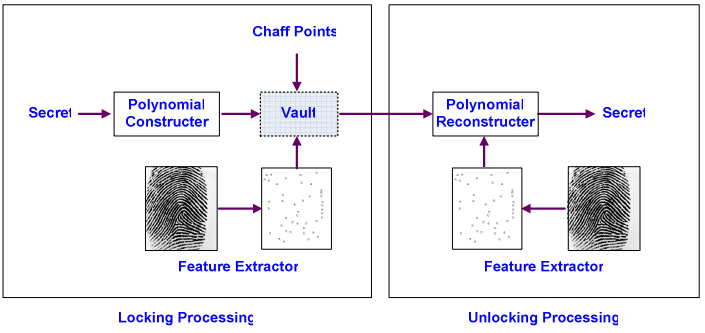


Fig. 1. Illustration of the Fuzzy Fingerprint Vault

3 Integration of Template Protection with Fingerprint Verification

This section explains a proposed approach to generate protected fingerprint templates and perform fingerprint verification with the protected templates. To explain our approach, we first describe the fuzzy vault in more detail. As explained in Section 2, we assume that Alice can place a secret value m in a vault and lock it using an unordered locking set $\boldsymbol{L}$. Bob, using an unordered unlocking set $\boldsymbol{U}$, can unlock the vault only if $\boldsymbol{U}$ overlaps with $\boldsymbol{L}$ to a great extent. The procedure for constructing the fuzzy vault is as follows: Secret value m is first encoded as the coefficients of some degree k polynomial in x over a finite field $GF(q)$. This polynomial $f(x)$ is now the secret to protect. The locking set $\boldsymbol{L}$ is a set of t values $l_i \in GF(q)$ making up the fuzzy encryption key, where $t > k$. The locked vault contains all the pairs $(l_i; f(l_i))$ and some large number of chaff points (α_j, β_j), where $f(\alpha_j) \neq \beta_j$. After adding the chaff points, the total number of items in the vault is r. In order to crack this system, an attacker must be able to separate the chaff points from the legitimate points in the vault. The difficulty of this operation is a function of the number of chaff points, among other things. A legitimate user should be able to unlock the vault if they can narrow the search space. In general, to successfully interpolate the polynomial, they have an unlocking set $\boldsymbol{U}$ of t elements such that $\boldsymbol{L} \cap \boldsymbol{U}$ contains at least $k + 1$ elements. The details of the fuzzy vault can be found in [10,15-16].

3.1 Generation of Protected Templates

As explained in Section 2, a fingerprint minutia represented by $m_i = (x_i, y_i, \theta_i, t_i)$ is composed of three elements such as *coordinates*, *angle*, and *type*. In typical fingerprint verification systems, minutiae are stored in the template file, and input minutiae are compared with the template minutiae after aligning them. In this typical fingerprint verification system, however, an attacker can reuse the template minutiae once he steals it. For protecting template minutiae from the attacker, we use the idea of the fuzzy vault scheme that stores a number of chaff minutiae generated randomly as well

as the real minutiae. That is, we consider the template(real and chaff minutiae) and the input minutiae as the locking and the unlocking set, respectively. Then, the attacker cannot reuse the fuzzy fingerprint vault directly without separating the real minutiae from the chaff minutiae even if he steals the fingerprint fuzzy vault. Note that, a legitimate user also should be able to separate the real minutiae to compute alignment parameters between the enrolled real minutiae and the input minutiae. However, the legitimate user cannot separate the real minutiae from the chaff minutiae, either. Therefore, we need a new alignment technique for the fuzzy fingerprint vault that does not need this separate operation, and we perform this alignment by modifying the geometric hashing technique[18]. Fig. 2 shows an example of a generated template consisting of real minutiae and chaff minutiae.

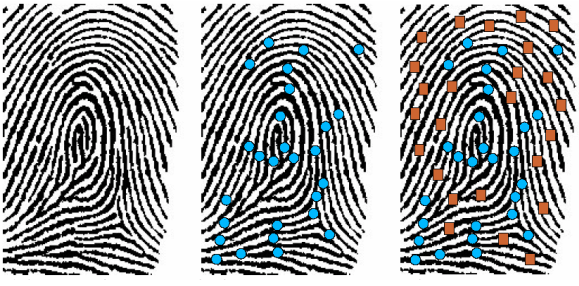


Fig. 2. Example of Generating Protected Templates[19]; (a) Example of a Fingerprint, (b) Example of a Typical Template(circles: real minutiae extracted from (a)), (c) Example of a Protected Template(circles: real minutiae, rectangles: chaff minutiae)

3.1.1 Selection of Chaff Minutiae

As in [16], we assume the number of the chaff minutiae as 200. However, it is challenging to perform fingerprint verification with the protected template added by this number of chaff minutiae. To increase the accuracy of our fuzzy fingerprint vault system, we determine the coordinates and the angles of the added chaff minutiae carefully. For example, if the coordinates and the angles of the added chaff minutiae are similar to those of a real minutia of a legitimate user, then the corresponding input minutia of the legitimate user can be aligned with the added chaff minutiae resulting in an incorrect alignment. Therefore, we first define acceptable margin Δd and $\Delta\theta$ for coordinate and angle, respectively. Since two minutiae within Δd and $\Delta\theta$ are considered as matched minutiae pair, the coordinates and the angles of the added chaff minutiae need to be determined outside the acceptable margin from any real minutiae for accurate fingerprint match. Additionally, newly added chaff minutiae need to consider the coordinates and the angles of the already added chaff minutiae in order not to reveal them as chaff minutiae. However, the type information of the chaff minutiae is selected randomly, because it is less important than other information.

3.1.2 Creation of Enrollment Minutiae Table

Let $m_i = (x_i, y_i, \theta_i, t_i)$ represent a minutia and $\boldsymbol{L} = \{m_i \mid 1 \le i \le r\}$ be a locking set including the real and chaff minutiae. In $\boldsymbol{L}$, the real and chaff minutiae can be

represented by $\boldsymbol{G} = \{m_i \mid 1 \le i \le n\}$ and $\boldsymbol{C} = \{m_i \mid n+1 \le i \le r\}$, respectively. Note that, the enrollment minutiae table is generated from $\boldsymbol{L}$.

In the *enrollment minutiae table generation stage*, an enrollment table is generated in such a way that no alignment is needed in the verification process for unlocking vault by using the geometric hashing technique. That is, alignment is pre-performed in the enrollment table generation stage. In verification process, direct comparisons without alignment are performed in 1:1 matching between the enrollment minutiae table and an input fingerprint in order to select the real minutiae($\boldsymbol{G}$) only. Each step in the enrollment minutiae table generation stage is explained in detail in the following.

1) Reference Point Selection Step

In reference point selection step, a minutia is selected as the first minutia from the set of enrollment minutiae($\boldsymbol{L}$). The first minutia is denoted by m_1 and the other remaining minutiae are denoted as $m_2, m_3, \ldots, m_n$. At this moment, the minutia, m_1, is called *basis*.

2) Minutiae Transform Step

In minutiae transform step, minutiae $m_2, m_3, \ldots, m_n$ are aligned with respect to the first minutia m_1 and quantized. Let $m_j(1)$ denote the transformed minutiae, *i.e.*, the result of the transform of the jth minutia with respect to m_1. Also, let T_1 be the set of transformed minutiae $m_j(1)$, *i.e.*, $T_1 = \{m_j(1) = (x_j(1), j_j(1), \theta_j(1), t_j(1)) \mid 1 < j \le r\}$, and T_1 is called the m_1-transformed minutiae set. (Eq. 1) represents the translation and rotation such that features $(x_1, y_1, \theta_1, t_1)$ of m_1 is translated and rotated into $(1,1,1,t_1)$. Let ${}_{TR}m_j(1)$ denote the minutia translated and rotated from the jth minutia with respect to m_1.

$$ {}_{TR}m_j(1) = \begin{pmatrix} {}_{TR}x_j(1) \\ {}_{TR}y_j(1) \\ {}_{TR}\theta_j(1) \\ {}_{TR}t_j(1) \end{pmatrix} = \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_j - x_1 \\ y_j - y_1 \\ \theta_j - \theta_1 \\ t_j \end{pmatrix}, where\ 1 < j \le r \qquad \text{(Eq. 1)} $$

$$ m_j(1) = \begin{pmatrix} x_j(1) \\ y_j(1) \\ \theta_j(1) \\ t_j(1) \end{pmatrix} = \begin{pmatrix} \lfloor {}_{TR}x_j(1)/\alpha + 0.5 \rfloor \\ \lfloor {}_{TR}y_j(1)/\alpha + 0.5 \rfloor \\ \lfloor {}_{TR}\theta_j(1)/\beta \rfloor \\ {}_{TR}t_j(1) \end{pmatrix} \qquad \text{(Eq. 2)} $$

To reduce the amount of information, quantization is required both in coordinates and angles. This quantization is summarized in (Eq. 2), where α is the quantization parameter for coordinates and β is the quantization parameter for angles. α is determined by the range of coordinates in the extraction stage in the enrollment process, whereas β is determined by the required precision in the verification process.

3) Repeat Step

Step 1) and step 2) are repeated for all the remaining minutiae. When step 1) and step 2) are completed for all the minutiae of the enrollment user, the *enrollment minutiae table* is generated completely.

Finally, the locked vault contains all the pairs $(x_i, y_i, \theta_i, t_i, f(x_i, y_i))$ and some large number of the chaff minutiae $(x_j, y_j, \theta_j, t_j, \beta_j)$, where $f(x_j, y_j) \neq \beta_j$.

3.2 Fingerprint Verification with the Protected Template

After the enrollment process, the verification process to separate the chaff minutiae(***C***) from the real minutiae(***G***) in the enrollment minutiae table should be performed. In the verification process, minutiae information(unlocking set ***U***) of a verification user is obtained and a table, called *verification table*, is generated according to the geometric characteristic of the minutiae. Then, the verification table is compared with the enrollment minutiae table, and the subset of real minutiae is finally selected. Note that, the verification table generation stage is performed in the same way as in the enrollment process.

In comparing the enrollment and verification minutiae tables, the transformed minutiae pairs with the same coordinates, the same angle, and the same type are determined. The minutiae pairs having the maximum number and the same basis are selected as the subset of real minutiae(***G***). Also, any additional alignment process is not needed because pre-alignment with each minutia is executed in the enrollment and verification minutiae table generation stage.

Note that, because of the noises and local deformation during acquisition, extracted minutiae from the same finger may have different coordinates and angles over each acquisition. To solve this problem, an adaptive elastic matching algorithm in which tolerance levels are determined according to the polar coordinates of the minutiae was proposed in [1]. In this paper, the coordinate plane is divided into several fields according to the distance from the origin. Each field has its own level of tolerance for x- and y-coordinates. The first field has tolerance level of [-3, 3] which means errors between -3 and 3 in x- or y-coordinate are tolerated. Two transformed minutiae in the first field are considered to have matching coordinates if their coordinates do not differ more than this error range. Tolerance level for angles is 22.5° for all transformed minutiae. Note that, this reduction of the search space required in a straightforward implementation of the geometric hashing can reduce the execution time significantly.

If the unlocking minutiae set ***U*** overlaps with the real minutiae of the locking minutiae set ***L*** in at least (polynomial degree $d + 1$) minutiae, for some combinations, the correct polynomial of degree d can be reconstructed. This represents the desired outcome when the locking and the unlocking minutiae sets are from the same finger. Since an attacker cannot separate the real minutiae from the chaff minutiae, he cannot reconstruct the correct polynomial without brute-force attacks.

4 Implementation Details and Experimental Results

For the purpose of evaluating of the accuracy of the proposed approach, a data set of 1,600 fingerprint images composed of four fingerprint images per one finger was collected from 400 individuals by using the optical fingerprint sensor[20]. The resolution of the sensor was 500dpi, and the size of captured fingerprint images was 248×292. As in [16], we selected the number of the chaff minutiae as 200, and they were generated by using a random number generator. The average number of minutiae in our data set is 36, and the average size of the enrollment minutiae table is 236×235.

Table 1. Experimental Results with Genuine and Imposter Sets

	Genuine	Imposter
Average # of Minutiae	36	36
Average # of Matched Minutiae	17	6
Average # of Matched Real Minutiae	16	2
Average # of Matched Chaff Minutiae	1	4

Table 2. Experimental Results for the Various Polynomials

	FRR	FAR
Degree-9 Polynomial	0.08	About 0
Degree-10 Polynomial	0.12	0
Degree-11 Polynomial	0.18	0
Degree-12 Polynomial	0.26	0

First, we evaluate the capability of separating the real minutiae from the chaff minutiae with the genuine and the imposter sets, respectively. As shown in Table 1, the average number of the matched minutiae is 16 and 1 for the real and the chaff minutiae in the genuine experiment, respectively. On the contrary, 2 and 4 minutiae are matched for the real and the chaff minutiae in the imposter experiment, respectively. Also, we can consider that the input minutiae are from a genuine if the unlocking set reconstructs the correct polynomial. If we use a 9-degree polynomial with 10 coefficients, 10 unique projections are required for unlocking the 9-degree polynomial. Based on our experiment, 4,391 of the 4,800 attempts were able to successfully unlock the vault in the genuine experiment. The remaining 409 query fingerprints selected fewer than the required number of the real minutiae(*i.e.,* less than 10), hence they were unable to unlock the vault. Also, the False Reject Rate(FRR) of the proposed approach is 0.081 and the Genuine Accept Rate is 0.915.

For evaluating the corresponding False Accept Rate(FAR), we tried to unlock the vaults with fingerprint minutiae of the imposter. Hence, we have 399 imposter unlocking attempts for a distinct finger, and only one fingerprint of the attempts could unlock the vault. Hence, experimental FAR is about 0%. Table 2 shows the experimental FRR and FAR for the various polynomials. As shown in Table 2, the FAR is 0% once using from 9-degree to 12-degree polynomials.

We can also analyze the security of the system mathematically as in [16]. Assume that we have an attacker who does not use real fingerprint data to unlock the vault; instead he tries to separate real minutiae from chaff minutiae in the vault using brute-force. If we use a 12-degree polynomial with 13 coefficients, attacker needs at least 13 minutiae to reconstruct the correct polynomial. The vault has 236 minutiae(36 of them are real, remaining 200 are chaff); hence there are a total of $C(236,13) \approx 8.0\times10^{20}$ combinations with 13 elements. Only $C(36,13) \approx 2.3\times10^{9}$ of these combinations will reveal the secret(*i.e.,* unlock the vault). Thus, it will need an average of 3.4×10^{11} (=C(230,9)/C(30,9)) evaluations for an attacker to crack the vault.

To guarantee for higher security, we are developing an approach to handle a more number of chaff points. Because adding the number of chaff points in restricted to the

size of the 2D hash table determined by a given fingerprint sensor, we are considering a 3D hash table. By using the 3D hash table, we can add more than 200 chaff points. However, it will take more execution time and memory space, and an efficient approach needs to be developed.

Finally, we compare the performance of the proposed approach and that of the Uludag's approach[16] as shown in Table 3. To compare two approaches, we show the performance with 8-degree polynomials that were selected in Uludag's experiment. The FRR and FAR of the proposed approach are 0.05 and 0.0016, respectively. As our fingerprint data set and variable parameters(*i.e.,* the number of the real minutiae) are different from ones used in Uludag's experiment, the reported FRR and FAR can be different with the environment of Uludag's experiment. Nevertheless, we should note that Uludag aligned two fingerprints manually while our proposed approach aligned automatically. To crack the vault, however, an attacker needs an average of 5.7×10^7 evaluations with our approach and 5.3×10^{10} evaluations with the Uludag's approach. This is the reason that our experiment uses more real minutiae(36 minutiae) than ones(18 minutiae) used in Uludag's experiment.

Table 3. Performance Comparison with Degree-8 Polynomials

	Uludag, et al.[16]	**Proposed Approach**
Alignment	Manual	Automatic
FRR	0.21	0.05
FAR	0	0.0016
Security	5.3×10^{10}	5.7×10^7

5 Conclusions

The use of biometrics in user authentication systems is very promising. However, without adequate security considerations, the compromise of such biometric data may make them useless for the user forever. In this paper, we proposed an approach to generate a non–invertible transformed version of the fingerprint data and to perform fingerprint verification with the protected fingerprint template. Based on the idea of the fuzzy vault, we generated a minutiae table that included the non–invertible transformed version of the enrolled fingerprint data, and then performed fingerprint verification on the minutiae table with the input fingerprint data.

To evaluate the effectiveness of our approach, we conducted experiments with actual fingerprint data. Based on the experimental results, our approach of using the minutiae table generated with the idea of the fuzzy vault can perform the fingerprint verification securely with automatic alignment.

Acknowledgement

This work was partly supported by the IT R&D program of MIC/IITA[2007-S-020-01, Development of Privacy Enhanced Biometric System] and by the MIC (Ministry of Information and Communication), Korea, under the Chung-Ang University

HNRC-ITRC(Home Network Research Center) support program supervised by the IITA(Institute of Information Technology Assessment

References

[1] Maltoni, D., et al.: Handbook of Fingerprint Recognition. Springer, Heidelberg (2003)
[2] Elliott, S.: Differentiation of Signature Traits vis-à-vis Mobile-and Table-Based Digitizers. ETRI Journal 26(6), 641–646
[3] Bolle, R., Connell, J., Ratha, N.: Biometric Perils and Patches. Pattern Recognition 35, 2727–2738 (2002)
[4] Prabhakar, S., Pankanti, S., Jain, A.: Biometric Recognition: Security and Privacy Concerns. IEEE Security and Privacy, 33–42 (2003)
[5] Uludag, U., et al.: Biometric Cryptosystems: Issues and Challenges. Proc. of IEEE 92(6), 948–960 (2004)
[6] Jeong, Y., Yoon, K., Ryou, J.: A Trusted Key Management Scheme for Digital Rights Management. ETRI Journal 27(1), 114–117 (2005)
[7] Matsumoto, T., et al.: Impact of Artificial Gummy Fingers on Fingerprint Systems. In: Proc. of SPIE, vol. 4677 (2002)
[8] Hill, C.: Risk of Masquerade arising from the Storage of Biometrics, BS Thesis - Australian National U., (2002)
[9] Schneier, B.: The Uses and Abuses of Biometrics. Communications of the ACM 42(8), 136 (1999)
[10] Juels, A., Sudan, M.: A Fuzzy Vault Scheme. In: Proc. of Symp. on Information Theory, p. 408 (2002)
[11] Moon, D., et al.: Implementation of the USB Token System for Fingerprint Verification. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 998–1005. Springer, Heidelberg (2003)
[12] Ratha, N., Connel, J., Bolle, R.: Enhancing Security and Privacy in Biometrics-based Authentication Systems. IBM Systems Journal 40(3), 614–634 (2001)
[13] Juels, A., Wattenberg, M.: A Fuzzy Commitment Scheme. In: Proc. of ACM Conf. on Computer and Comm. Security, pp. 28–36 (1999)
[14] Dodis, Y., Reyzin, L., Smith, A.: Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Heidelberg (2004)
[15] Clancy, T., Kiyavash, N., Lin, D.: Secure Smartcard-based Fingerprint Authentication. In: Proc. of ACM SIGMM Multim., Biom. Met. & App., pp. 45–52 (2003)
[16] Uludag, U., Pankanti, S., Jain, A.: Fuzzy Vault for Fingerprints. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 310–319. Springer, Heidelberg (2005)
[17] Barral, C., et al.: Externalized Fingerprint Matching. In: Zhang, D., Jain, A.K. (eds.) ICBA 2004. LNCS, vol. 3072, pp. 309–315. Springer, Heidelberg (2004)
[18] Wolfson, H., Rigoutsos, I.: Geometric Hashing: an Overview. IEEE Computational Science and Engineering 4, 10–21 (1997)
[19] Adler, A.: Vulnerabilities in Biometric Encryption Systems. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 1100–1109. Springer, Heidelberg (2005)
[20] Ahn, D., et al.: Specification of ETRI Fingerprint Database(in Korean), Technical Report – ETRI (2002)

Design and Application of Optimal Path Service System on Multi-level Road Network

Yumin Chen[1], Jianya Gong[2], and Chenchen wu[2]

[1] School of Resource and Environment Science, Wuhan University,
Luoyustr.129, 430079 Wuhan, China
[2] State Key Laboratory for Information Engineering in Surveying, Mapping
And Remote Sensing, Wuhan University,
Luoyustr.129, 430079 Wuhan, China
{ycymlucky, cola005, gongjianya}@hotmail.com

Abstract. Classical algorithms of optimal path analysis are not efficient enough when handling large numbers of road network nodes. With the light of web service and internet, this problem has a breakthrough development. They provide technique support for dynamical building of multi-level road network data model and realization of the web service of optimal path analysis. This paper designs and implements a service system on multi-level road network. The result of this application proves that the optimal path service on multi-level road network is feasible and efficient. It can well solve the optimal path analysis problem in large area.

Keywords: Multi-level road network, the optimal path, web service system, GML.

1 Introduction

Classical optimal path algorithm can be used to solve ordinary optimal path search problem. However, when the network contains too many nodes, data will exceed the limitation that a geometry network can handle. According to this, classical optimal path algorithm often can not get results in effective time. So it is hard to satisfy the application request. As extension of urban area become larger, the city road networks become more complex and huger. Usually, a typical city road network is very huge, including thousands upon thousands of nodes. If several cities are considered, even countrywide or larger area, the classical algorithm's ability is not equal to its ambition.

To solve this problem effectively, when a mass of data is given, distributed multi-level road network data is considered. It builds pyramid-liked data model, calculates the optimal path analysis in multiple layers. This method can reduce amount of calculation, cut down calculation time. Emergence of Web Service technology and development of Internet network technology provided necessary preparation for data integration of a distributed multi-level network.

At present, many scholars already contribute their research on such system and framework of Web Service. And it is used in lots of industries gradually. There are

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1152 – 1164, 2007.
© Springer-Verlag Berlin Heidelberg 2007

some relative researches in ITS field, such as vehicle optimal path search service based on Internet methodology (Hassan A.Karim, Ratchata Peachavanish, 2004), static and dynamic traffic information service model (Hong K.Lo, W.Y.Szeto, 2004) , and so on. Distributed road network data is collected from different data sources, regions, scales, GIS platforms can be integrated through Web Service technology. So it provides excellent technique support for a dynamical building of pyramid-liked multi-level road network data model and an optimal path service.

This paper researches on design of service application system, establishment of server and client, and workflow expression in details. And it proposes an advanced design plan. The application instances indicate that the optimal path service system of multi-level road network road network is feasible, and provides an efficient path analysis. It can solve the integration and application problems of large region road network path analysis.

2 System Design

Optimal path service system on Multi-level road network based on Web Service technology uses Web service measure to retrieve data service and functional service. It implements optimal path analysis hierarchically, through building pyramid-liked multi-level road network data model. As we all know, the multi-level road network data of the system comes from data-sharing service provided by different regions, scales, GIS platforms. WMS, WFS and WCS standard provided by OGC definitely define a serious service interfaces to satisfy requirement of data-sharing service. Forcing by this development, now majority of GIS software, ArcGIS, MapInfo, GeoStar software platform, gradually support for WMS, WFS, WCS standard interface. When these services registered in UDDI Registry, services needed could be searched through UDDI registry. And relative road network data can be retrieved. On this basis, pyramid-liked multi-level road network data model is built. The identification ways of weight value information are decided by users.

For multi-level road network data model been built, use optimal path algorithm of multi-level road network to do the optimal path analysis. When analyze each level of road network, use dynamically build topology relation method to ensure topological coherence. This system Saves final analysis result in uniform GML application criterion as GML documents, and then transfers them to client. Client interprets GML documents and display visualization result. Client provides relative user interface operation functions at the same time. The service chain process of optimal path service system of multi-level road network is shown in fig.1.

Data-sharing service is provided through WMS and WFS interface that are supported by the existing software ArcIMS, MapInfo, and the WMS, WFS and WCS interface that supported by indigenous GIS software GeoStar and GeoSurf components. GeoStar and GeoSurf components respectively carry out the data-sharing interface of OGC on Microsoft Visual Studio.Net and Java platform. Data format conversion and data projection conversion service are network functions developed by Microsoft Visual Studio.Net platform. It provides functional operation for building data model and path analysis. By choosing different software platforms and development

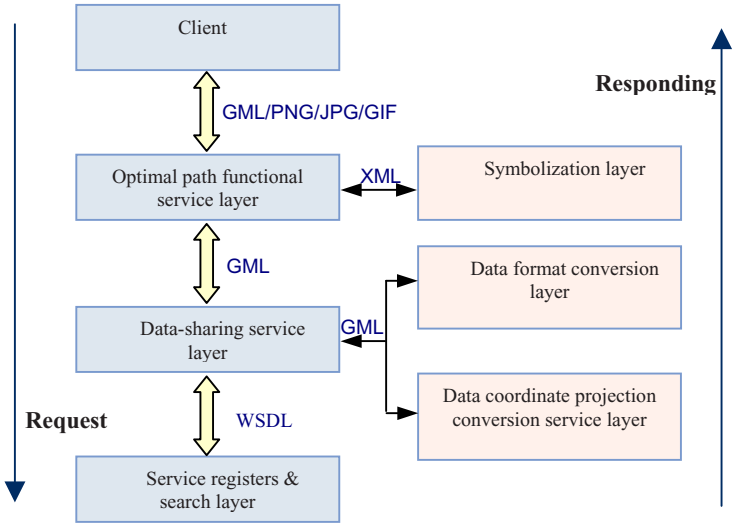


Fig. 1. Multi-level road network optimal path service chain

languages, it was certified that multi-level road network optimal path service can well integrate distributed road network data, and can realize platform-cross interoperability of path analysis.

3 System Framework

The system is divided into two parts: one is system server design; the other is system client design. The details of the framework are as follows:

3.1 System Server Design

System server provides various kinds of data-sharing services, function-sharing services, and other functions required by server. It mainly includes server registration and search layer, data-sharing service layer, data format conversion service layer, data coordinates projection conversion service layer, optimal path functional service layer, and so on.

3.1.1 Data-Sharing Service

GIS data-sharing and operability are the cores of new generation GIS system based on web Service. The data-sharing service standards of OGC, like web map service (WMS), web feature service (WFS) and web coverage service (WCS), are gradually accepted by major GIS software developers. And they provide responding support interface in respective software. This brings benefits for data-sharing service of different GIS software platforms. The system search registered services through UDDI registry. It searches distributed road network data service which is needed by system and build multi-level road network data model. Criterion measured up WMS, WFS standard interfaces are provided through service. The output information of WMS

service is map in JPG/PNG/GIF image forms. The output information of WFS is GML document which satisfies distributed road network GML application schema. It is shown in fig.2.

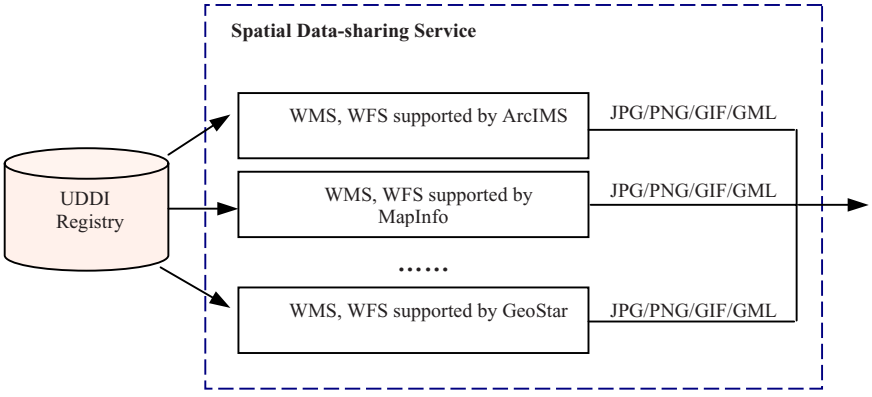


Fig. 2. Spatial data-sharing service

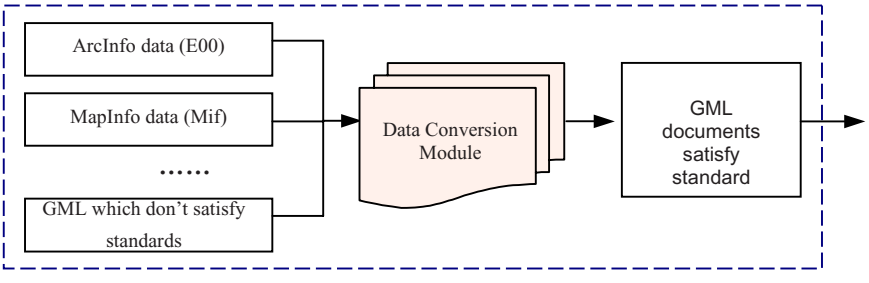


Fig. 3. Spatial data conversion service

GML documents outputted by WFS service of different GIS software developers like ArcIMS, MapInfo, may don't satisfy the request of distributed road network GML application schema. So it needs to convert data to ensure data format coherence. When projection is processing, coordinates information of road network are different. So data projection conversion and coordinate conversion are necessary. It enables data from various data sources can integrate then build the multi-level road network data model, as shown in fig.3. Spatial data-sharing service framework of server is shown in fig.4.

3.1.2 Function-Sharing Service

Distributed multi-level road network data can be retrieved through spatial data-sharing service. It provides necessary data source for optimal path functional service. Optimal path service system on multi-level road network mainly includes building of multi-level road network, optimal path calculation of multi-level road network, and

symbolization of optimal path. The functional service is also realized by Web Service technology. It supports SOAP protocol. Server data transmission standard gratify the WMS and WFS standard of OGC. The output information of WMS service is JPG/PNG/GIF map image. The output information of WFS is GML document satisfies distributed road network GML application schema. Due to the network components characteristic of Web Service technology, each function operation of optimal path service is defined following the interfaces. It is written as responding Web Service standard interface. So it can be easily integrated into the system. Optimal path functional service on multi-level road network is revealed in fig.5.

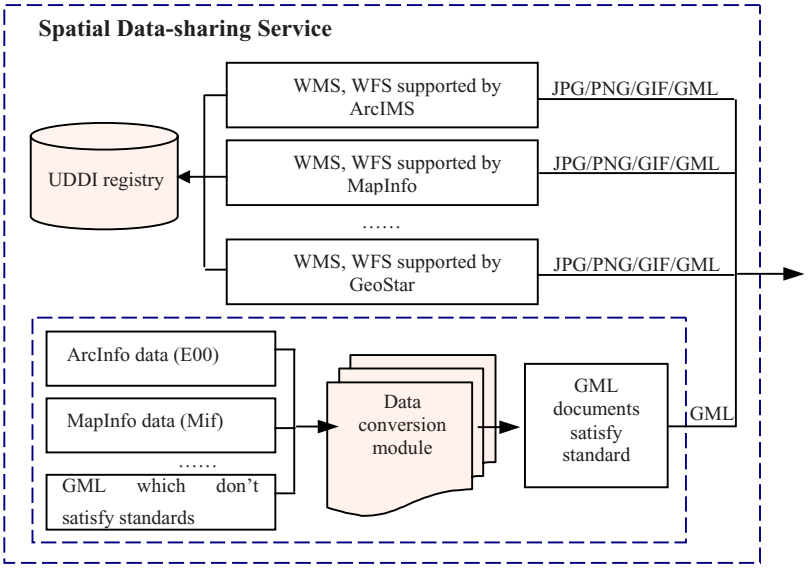


Fig. 4. Spatial data-sharing service framework

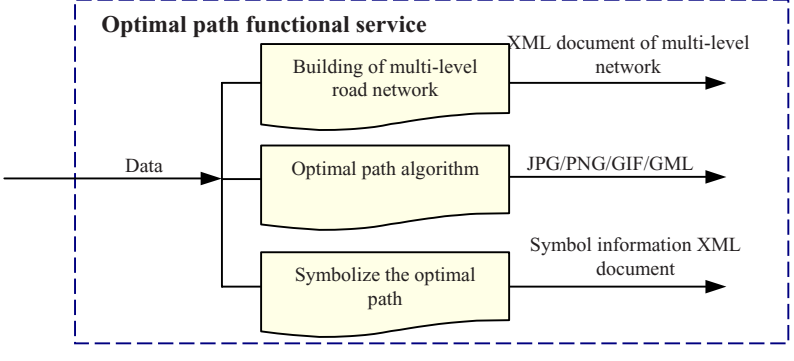


Fig. 5. Optimal path function-sharing service on multi-level network

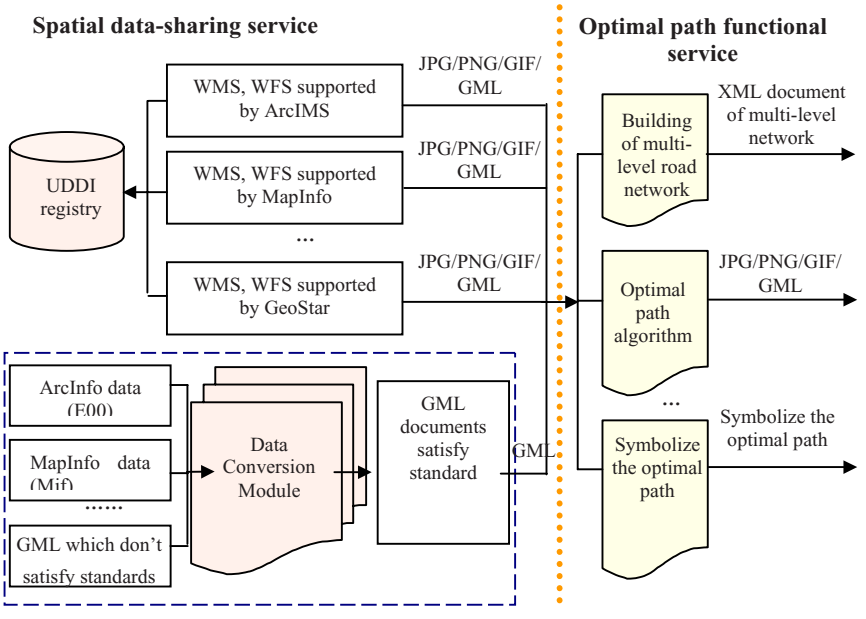


Fig. 6. System server framework design

To summarize the above data-sharing services and function-sharing services, the server framework of optimal path service system on multi-level road network is shown in fig.6.

3.2 System Client Design

Server provides integration of optimal path data-sharing services and function-sharing services on multi-level road network. Client transfers these services to fulfill all kinds of functional operations expected by users. It seems that server makes a series network components based on Web Service, then according to demand of users, client uses appropriate network components to exploit the secondary development, and set down the final user interface.

When an operational request is sent by users, client redelivers this request to server. And then server carries out a corresponding Web Service to respond this request. And it generates map images/GML documents of JPG/PNG/GIF result, and then returns them to client. Client displays map images or interpret GML documents to retrieve finally display results.

Client has to do works as follows: (1) Through user operation interface, client interpret user request, transfer it to responding server Web Service to handle user request; (2) When server Web Service generates a responding GML document, client interpret this GML document, and generate a visualized graphic result; (3) For graphic display, client has simple zoom in, zoom out, and cruise functions, to satisfy client basic functional operation request; (4) Completing client interface needed by optimal path analysis on multi-level road network.

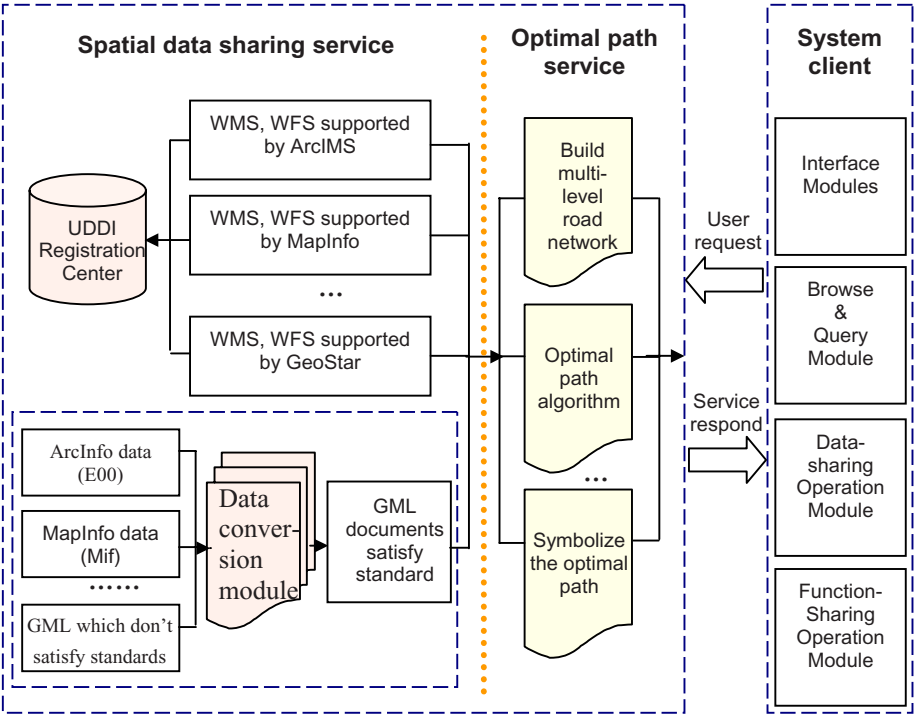


Fig. 7. Design of system server end and client end

Similar to server, the client can also be realized by Java platform or .NET platform. Client developed whether by Java platform or .NET platform, can easily call for Web Services. Distributed multi-level road network studied in this paper chose .NET platform to program the client. Design of system server and client is shown in fig.7.

4 Application Example

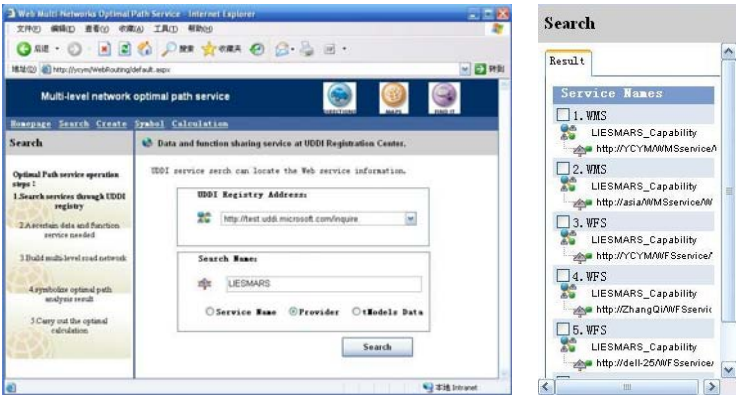
The final experiment application system is accomplished according to the system general design plan described above. According to operation process of optimal path service on multi-level road network, the system includes search and find of services, building of multi-level road network, symbolized setting of analysis result, optimal path calculation.

4.1 Search and Find Service

The first step of optimal path service operation on multi-level road network is to ascertain necessary data-sharing service and function-sharing service. The search work is carried out through UDDI registry and this experiment system is designed to search through Microsoft UDDI registry. It uses Third Party Taxonomy method based on scale information, to register and search distributed road network data services in

UDDI registry. After service providers register the data service into UDDI registry, the system can search all kinds of required services through the registry.

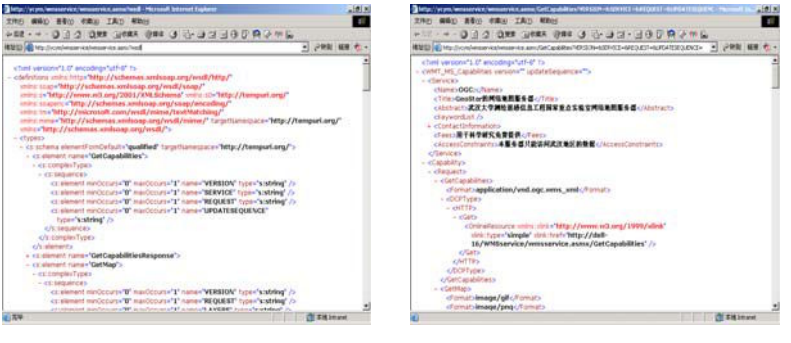
Fig.8 shows UDDI registry services search operation, fig.(a) is search page, fig.(b) is a list of searched service information. Services needed are searched through UDDI registry address and search name. Services matched requirements are listed in the result information list. The list information includes service name, GetCapabilities operation of service, and WSDL address of service. Microsoft.Uddi.Sdk.dll component is used in practice to facilitate the service search operation.



(a) Services search of UDDI registry (b) Search result

Fig. 8. Services search operation of UDDI registry

Service description information and metadata information can be retrieved through WSDL address and GetCapabilities operation of service. Fig.9 shows WSDL information and GetCapabilities operation information provided by WMS service http://ycym/wmsservice/wmsservice.asmx.



(a) WSDL information of services (b) GetCapabilities information of services

Fig. 9. WSDL and GetCapabilities of search service

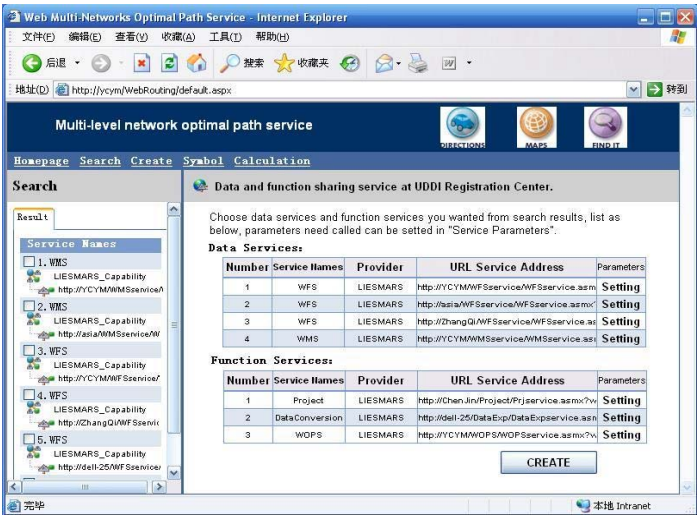


Fig. 10. Choose data service and functional service that needed

After check the service information, the user could confirm service he desires. Fig.10 shows data services and functional services which are been chosen.

4.2 Build Multi-level Road Network

Distributed road network data is come from GML documents of WFS services provide by different GIS software platforms. So WFS service is chosen from data-sharing services to realize the building of multi-level road network. Fig.11 shows three WFS services from different servers, and GML document of each WFS service follows the uniform data projection, data format. If data formats are not coherent, data format conversion and data projection coordinate conversion are necessary.

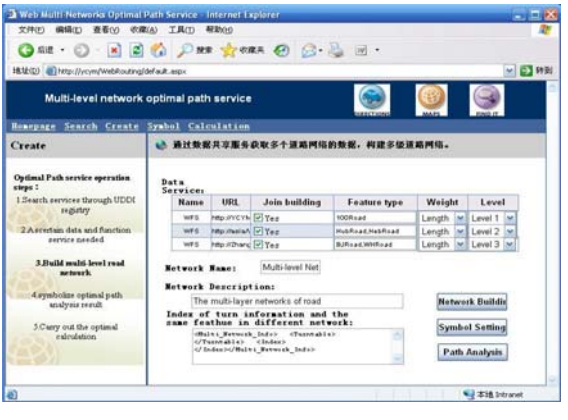


Fig. 11. Building of multi-level network

Following items are set respectively: whether WFS service participates in building multi-level road network, road network feature class, the weight value, and the grade number of multi-level road network. In addition, the multi-level road network name, description information, turn and index information, and other parameters are set. They are the basis to build multi-level road network. Multi-level road network can display XML document of road network building information after being built. This experiment chooses national 1:4,000,000 road network data, 1:1,000,000 road network data of Hubei province, Hebei province and Beijing area to build three-level road network.

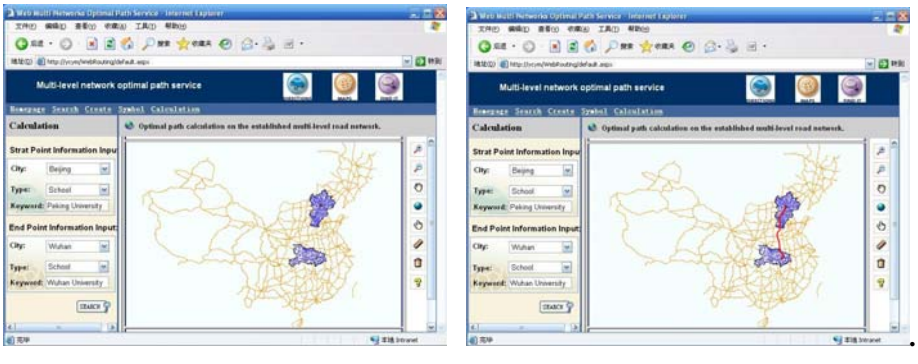
4.3 Symbolization Setting of Analysis Result

The symbolized display of optimal path analysis is set. The optimal path analysis result is linear feature. So the setting of symbol can complete through choosing user-defined simple symbol setting, such as symbol color, style and width. It can be set through symbol ID in symbol library as well.

4.4 Optimal Path Calculation on Multi-level Road Network

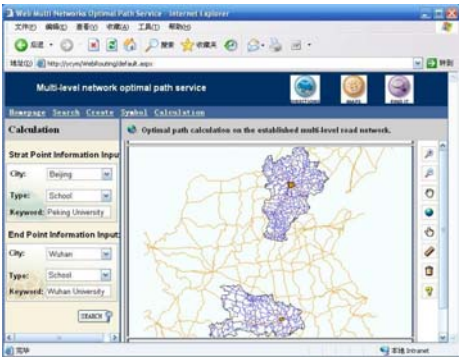
After building and symbolization setting of multi-level network, the optimal path calculation can be processed. The optimal path calculation is carried out by inputting information of start point and end point, as shown in fig.12 (a to h). They respectively describe three-level road network overlapped map and three-level road network optimal path analysis result map, overlapped map and optimal path analysis result map that zoomed in, Beijing area three-level road network overlapped map and Beijing area optimal path analysis result, Wuhan area three-level road network overlapped map and Wuhan area optimal path analysis result.

The time cost by data retrieving, calculation process of optimal path analysis service in multi-level road network is shown in table.1. It shows that the total data size is about 10M, including national road network, road network of Hebei (including Beijing), Hubei, and road network of Beijing and Wuhan. When dealing each level within road network, during the increase of data volume, time used for data retrieving, analysis, display, and the optimal path computation will accordingly increase. And it

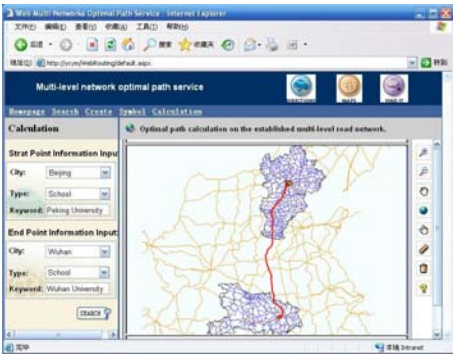


(a) Three-level road network overlapped map (b) Three-level road network

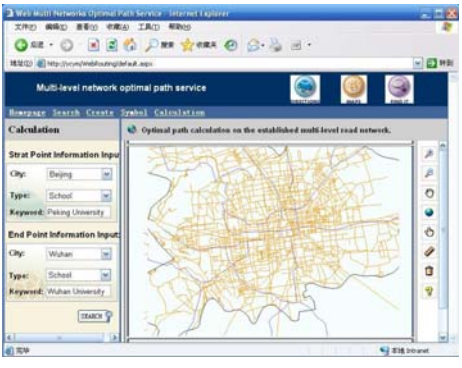
Fig. 12. Optimal path calculation of multi-level network



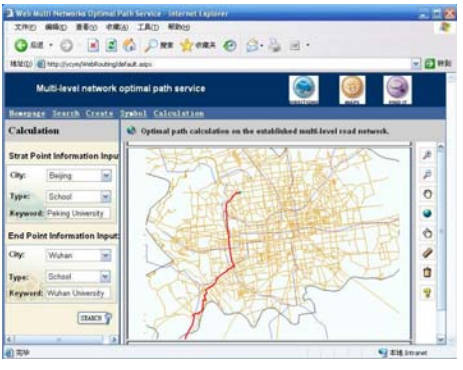
(c) Zoomed in overlapped map of Three-level road network



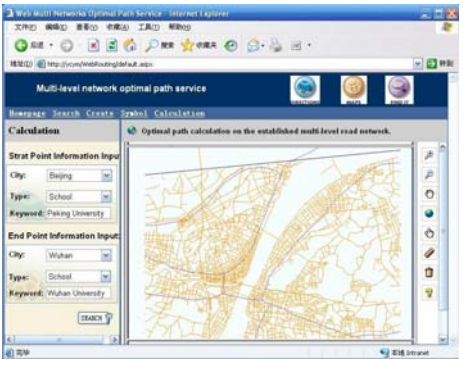
(d) Zoomed in map of Optimal path analysis result



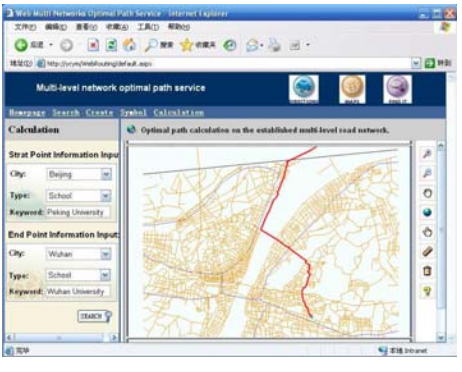
(e) Overlapped map of Beijing area three-level road network



(f) Beijing area optimal path analysis result



(g) Overlapped map of Wuhan area three-level road network



(h) Wuhan area optimal path analysis result

Fig. 12. *(continued)*

Table 1. Time table of optimal path analysis on multi-level road network

Level	Data source	Data (KB)	Time of Data retrieving (ms)	Time of Data Interpretation (ms)	Time for building dynamic topology (ms)	Time of optimal path analysis (ms)	Total time of data retrieving and interpretation (ms)
Level 1	1:4,000,000 National Road Network	853	8,953	843	156	94	227,857
Level 2	1:1,000,000 Hebei Road Network	1,380	15,766	1,562	297	47	
	1:1,000,000 Hubei Road Network	694	6,578	609	219	16	Total time of path analysis (ms)
Level 3	1:10,000 Beijing Road Network	3,383	84,406	10,140	547	172	2,407
	1:10,000 Wuhan Road Network	3,873	85,594	12,406	609	250	

can be found that retrieving data through data sharing network services costs 227,857 milliseconds. Much more than 2,407 milliseconds are taken by multi-level road network path analysis. The time spent on building topologic information is a little more than optimal path analysis, but both of which are not comparable to data retrieving and data interpretation. So time consumption is mainly used in the data network transmission and interpretation. And the optimal path analysis algorithm on multi-level network has higher efficiency than others, and it is satisfactory for most demand of multi-level network path analysis service.

5 Conclusion

This paper designed and developed an optimal path service system on multi-level road network as a test of the idea mentioned. This paper mainly introduces the general design of this service application system, technology measure and the result. The service application proves that the optimal path service on multi-level road network is feasible and efficient to deal with large area optimal path analysis problem.

Acknowledgement

This research is funded by National Nature Science Foundation of China (project NO. 40601082). The authors would like to thank NSFC for this support.

References

1. Molin, E.J.E., Timmermans, H.J.P.: Traveler expectations and willingness-to-pay for Web-enabled public transport information services. Transportation Research Part. C: Emerging Technologies 14(2), 57–67 (2006)
2. Karacapilidis, Lazanas, N., Alexis.: On the development of a web-based system for transportation services. Information Sciences 176(13), 1801–1828 (2006)
3. Taha, O., Matthieu, S.: An integrative framework for traffic telematics web services. In: Proceedings of the IASTED International Multi-Conference on Applied Informatics, Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks, as part of the 23rd IASTED International Multi-Conference on Applied Informatics, pp. 375–380 (2005)
4. Karimi, H.A., Peachavanish, R.: Finding Optimal Bus Service Routes: Internet-Based Methodology to Serve Transit Patrons. Journal of Computing in Civil Engineering , 83–91 (2004)
5. Lo, H.K., Szeto, W.Y.: Modeling advanced traveler information services: static versus dynamic paradigms. Transportation Research Part. B 38, 495–515 (2004)
6. Arampatzis, G., Kiranoudis, C.T., Scaloubacas, P.: A GIS-based decision support system for planning urban transportation policies. European Journal of Operational Research 152, 465–475 (2004)
7. Demirel, H.: A dynamic multi-dimensional conceptual data model for transportation applications. ISPRS Journal of Photogrammetry & Remote Sensing 58, 301–314 (2004)
8. Jha, M.K., Schonfeld, P.: A highway alignment optimization model using geographic information systems. Transportation Research Part. A 38, 455–481 (2004)
9. Yumin, C., Jianya, G., Wenjue, J., Qi, Z.: A Model for Web Spatial Data Visualization Based on the Oracle Spatial Database. In: Fourth International Conference on Virtual Reality and its Application in Industry. vol. 5444, pp. 474–477. SPIE publisher (2004)
10. Yin, H., Wong, S.C., Xu, J., Wong, C.K.: Urban traffic flow prediction using a fuzzy-neural approach. Transportation Research Part. C 10, 85–95 (2002)

Spatio-temporal Similarity Measure Algorithm for Moving Objects on Spatial Networks

Jae-Woo Chang, Rabindra Bista, Young-Chang Kim, and Yong-Ki Kim

Dept. of Computer Eng., Chonbuk National Univ.,
Chonju, Chonbuk 561-756, South Korea
jwchang@chonbuk.ac.kr,
{rabindra,yckim,ykkim}@dblab.chonbuk.ac.kr

Abstract. In this paper, we propose a new spatio-temporal similarity measure to compute spatio-temporal relevance between two trajectories of moving objects on road networks, which is known as spatio-temporal distance (STDist). In addition, we provide a similar trajectory search algorithm to retrieve similar trajectories based on the proposed measure i.e., a combination of both spatial and temporal properties with respect to the motion of a given query trajectory. The performance evaluation shows that our approach outperforms the existing work in terms of searching similar trajectories of moving objects on road networks.

Keywords: Spatio-temporal Similarity, Algorithm, Addition-based Approach, Multiplication-based Approach.

1 Introduction

Similarity search for moving objects trajectories has attracted a lot of research work. Trajectory data are common in many diverse applications, such as video tracking [1], motion capturing [2], data analyzing tasks [3], [4], tracking animal migrations, cyclone trajectories [5], interactive generation of motions [6], discovering of subtle patterns during cellular mitoses [7]. With the spread of mobile computing, research on efficiently handling moving objects has become important [8], where their movement is represented as a trajectory including a set of line segments in (x, y, t) space.

The concept of similarity search of moving object trajectories is an important one useful for analyzing the behavior of moving objects on road networks. Similarity measurement allows for the retrieval of similar trajectories and for eventual discovery of the knowledge from patterns of the similar trajectories. For example, we can make marketing strategies by analyzing the trajectories of cars on road networks. Hence, there is a need to define a measure to determine similarity among the trajectories of moving objects. Although, much research has been done with regard to locating similar trajectories of moving objects in Euclidean space, there has been very little research regarding moving objects in road network space. For most of real applications, we are interested in moving objects in road network space rather than in Euclidean space. The methods used to search for similar trajectories in Euclidean space [9]-[16] are no longer useful for road network space, due to its different properties. The

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1165–1178, 2007.
© Springer-Verlag Berlin Heidelberg 2007

existing methods have the following drawbacks. First, Euclidean distance, which they are based on, is no longer valid for road network space, where the distance is limited to the space on the roads. Since measuring similar trajectories is highly dependent on the definition of distance, the similarity measures defined for Euclidean space are inappropriate for road network space. Second, the existing methods do not fully exploit the spatio-temporal properties of trajectories, and most of them only consider spatial similarities. For example, two trajectories passing through the same area at different times are considered similar, even though they are not similar in spatio-temporal domains.

In order to compare similarity of moving object trajectories in spatial networks, we propose three new similarity measures: i) spatial similarity measure to compute spatial distance between two trajectories, ii) temporal similarity measure to compute temporal distance between two trajectories, and iii) spatio-temporal similarity measure to compute spatio-temporal aspects of the moving object trajectories. Our spatial distance function calculates real road network distance unlike Euclidean distance of the existing researches. Based on the three measures, we propose a similar trajectory search algorithm to efficiently retrieve similar trajectories. The rest of the paper is organized as follows. In Section 2, the related work is presented briefly. Similarity measures and a new similar trajectory search algorithm based on them are presented in Section 3. In Section 4, we present the performance analysis of the proposed similar trajectory search algorithm. Finally, we conclude the paper with future work in Section 5.

2 Related Work

The problem of discovering similar trajectories received increasing attention in recent years and this motivated the development of various algorithms. The focus in this area lies primarily on the development of distance functions for similarity measurement as well as the context of the applications to be dealt. Because our concern is particularly dealing with the searching similar trajectories of moving objects in a spatial network, like a road network, we will briefly discuss related work on similar trajectories search in this section. Firstly, Vlachos et al [9] presented an investigation for analysis of spatio-temporal trajectories for mobile objects where data contain a great amount of outliers, based on the Longest Common Subsequence (LCSS), in conjunction with a sigmoid matching function to increase the performance than those of Euclidean and Time Warping Distance. Secondly, Yanagisawa et al [10] focused on the extraction of the individual moving patterns of each object from the trajectories considering both time and location. Their approach is based on the shape similarity between lines to retrieve required objects. Thirdly, Shim and Chang [11] dealt with similar sub-trajectory retrieval for moving objects, but their scheme is not appropriate to address the problems of road networks because of being based on Euclidean distance. Fourthly, Lin et al [12] focused on the spatial shapes and compared spatial shapes of moving object trajectories by developing algorithms for evaluating OWD (one way distance) in both continuous and discrete cases of the trajectories for similarity search. Two problems of this approach are: i) their search scheme is time independent and ii) OWD is based on Euclidean distance which can not calculate real road network distance to compare two trajectories. Fifthly, Zeinalipour-Yazti et al [13] proposed

distributed spatio-temporal similarity search based on the LCSS distance measure. Their scheme performs local computations of partial lower and upper bounds at each cell and then combines these partial results to give upper and lower bounds. Since their scheme is based on Euclidean distance, it can not address spatial networks. In addition, work of Vlachos et al [14], Sakurai et al [15], and Chen et al [16] can not address the problems of road network applications as they are based on the Euclidean distance. Finally, to the best of our knowledge, there exists only one research, which has been done by Hwang et al [17], to retrieve similar trajectories of moving objects on road networks based on both spatial and temporal aspects. The algorithm for searching similar trajectories consists of two steps, the filtering step based on spatial similarity and the refining step to search for similar trajectories based on temporal distance. During the filtering step, it ignores trajectories like ones passing through say, nine POIs (points of interest) out of ten POIs of a query but it is exactly similar to the query temporally. These trajectories are must to the applications like finding suitable route during rush hours. Hence the final effect of their approach [17] is that a candidate set fetched for a given query is devoid of all relevant data trajectories. Furthermore, their approach can not compare two trajectories of different lengths. These are the two main motivations behind our work.

3 Similarity Measures

To overcome the problems of the existing methods above mentioned in related work, we propose a method that exploits the property of spatial networks and considers time as an important domain to search similar trajectories from a relevant candidate set of moving objects trajectories. In this section, we will describe our three similarity measures along with a similar trajectory search algorithm based on them.

3.1 Spatial Similarity Measure

Spatial similarity of trajectories can be defined in terms of spatial distance (SD), a real distance between two trajectories in spatial network. So a data trajectory can be said to be spatially more similar to a query trajectory when a spatial distance between them is less than the spatial distance between the query trajectory and any other trajectory. For this, we define a new spatial similarity measure that computes spatial distance (non Euclidian distance) between two trajectories on road networks.

3.1.1 Spatial Similarity (SS)

We define spatial similarity (SS) in terms of SD. Assume that there are two trajectories say a query trajectory Q and a data trajectory R, where Q has a set of edges, Q = {q1, q2, q3,…,q_n} with the start time Qst and the end time Qet, and R has another set of edges, R = {r1, r2, r3,….,r_m} with the start time Rst and the end time Ret.

Definition 1. *Spatial distance between an edge pair Dedge(qe, re) from two different trajectories (Q, R) on road networks can be defined by averaging the distance between the start nodes of qe and re (i.e. SNodes) and the distance between the end nodes of qe and re (i.e. ENodes).*

$$D_{edge}(q_e, r_e) = \frac{D_n(q_e SNode, r_e SNode) + D_n(q_e ENode, r_e ENode)}{2}, \quad (1)$$

where D_n(n1,n2) is a spatial network distance between two nodes, n1 and n2.

Definition 2. *For given two trajectories Q and R, a set of matching edge pairs between them can be defined as MEdgePair(Q,R) where a member of MEdgePair(Q,R) can be represented as an edge qe from Q, another edge re from R, and a spatial distance Dedge(qe, re) between them. Here, Dedge(qe, re) is the lowest one among all the distances from qe to the rest of the edges of R.*

$$MEdgePair(Q, R) = \{(qe, re, Dedge(qe, re))\}, \quad 1 \leq e \leq m \quad (2)$$

where e= number of matching edge pairs between Q and R

Definition 3. *Spatial Similarity (SS) between two trajectories Q and R can be defined as the average spatial network distance of MEdgePair(Q,R).*

$$SS(Q, R) = \frac{1}{m}\left[\sum_{k=1}^{m} D_{edge}(q_k, r_k)\right], \quad (3)$$

where m= number of matching edge pairs between Q and R

3.1.2 Spatial Similarity Measure Algorithm

This algorithm shows spatial relationship between two trajectories of moving objects on road networks. For given two trajectories, it generates a sequence of matching edge pairs with spatial distance between two edges of each edge pair and computes spatial similarity of the trajectories. Algorithm 1 presents spatial similarity measure. The step 1 of the measure finds matching edge pairs for every query edge from R where the networks distance between the edges of each matching pair is the lowest one. If there remain some unselected edges of R, then the step 2 finds a matching edge pair for every unselected edge of R from the edges of Q. The step 3 adds spatial distances of all matching edge pairs and averages them to compute the spatial similarity measure between Q and R.

Spatial Similarity Measure Algorithm:
Input: A query trajectory Q and a moving object trajectory R
Output: Spatial Distance (SD) between Q and R

1. *for each edge qe of Q, find a matching edge re from R based on lowest network distance*
2. *for each unselected edge re of R, find a matching edge qe from Q like step 1*
3. *Add network distances of m matching edge pairs and average it*

Algorithm 1. Spatial similarity measure

3.2 Temporal Similarity Measure

Time domain plays an important role in searching similar trajectories of moving objects. We define temporal similarity between two trajectories in terms of temporal distance (TD) which is the time difference between time values of two trajectories. A data trajectory is said to be temporally more similar to a query trajectory when the temporal distance between them is less than temporal distance between the query trajectory and rest of data trajectories. We develop a novel approach to calculate a temporal distance of trajectories by using a time triplet consisting of time range of a day (R), day of a week (D) and week (W). We first calculate temporal distance values of R, D and W separately, and then we add all of them to get the final temporal distance between a given query trajectory and a data trajectory.

3.2.1 Temporal Similarity Based on Time Range of a Day (R)

In a road network, there can be different road conditions for running vehicles on the different time periods of a day. The patterns of trajectories of moving objects vary according to the time duration of the day. Focusing on this type of scenario of road networks, to define time range distance (RD), we first divide a single day into seven time ranges as shown in Table 1 and then we arrange them into four groups based on their similar traffic conditions as shown in Table 2. We assume that a query trajectory Q has the start-time Qst and the end-time Qet and a data trajectory T has the start-time Tst and the end-time Tet.

Table 1. Different time ranges of a day

S.No.	Range(R)	Description	From	To
1.	MR	Morning Rush Hour	7:00 AM	9:00 AM
2.	M	Morning	9:00 AM	12:00 PM
3.	L	Lunch Time	12:00 PM	2:00 PM
4.	A	Afternoon	2:00 PM	5:00 PM
5.	ER	Evening Rush Hour	5:00 PM	7:00 PM
6.	LE	Late Evening	7:00 PM	10:00 PM
7.	N	Night(Sleeping Time)	10:00 PM	7:00 AM

Table 2. Group of time ranges

Group	Member/s
Group 1	MR, ER
Group 2	L
Group 3	M, A, LE
Group 4	N

Definition 4. *If Q and T have motions in the same time range, then range distance (RD) between them equals to zero (0). If Q and T have motion in different time range but in the same group, then RD between them equals to 1.*

Definition 5. If Q and T have motions in different groups, then RD between them is given by

$$| Groupi - Groupj |*2 + 1, \quad where\ i, j = 1, 2, 3\ and\ 4. \tag{4}$$

For instance, if Q has motion in the range MR i.e. Group1 and that of T is in LE i.e. Group3, then RD between them is calculates as given below:

$$Group\ difference = | Group1 - Group3 | = |1-3| = 2. \quad So,\ RD = 2*2+1 = 5$$

Definition 6. Suppose that the motions Q and T start at one time range and end at another time range. Trajectory range distance (TRD) between them can be given by the average of start time range distance and end time range distance.

$$TRD = (RD(Qst, Tst) + RD(Qet, Tet)) /2 \tag{5}$$

For example, the motion of Q starts in MR and ends in M while that of T starts in ER and ends in LE. TRD between Q and T is calculated as TRD= (1+1)/2 = 1, where the former 1 is the distance between MR and ER and the latter 1 is the distance between M and LE.

3.2.2 Temporal Similarity Based on Days of a Week (D)

During a week, the influx of moving objects (vehicles) varies from working days (Monday to Friday) to weekend (Saturday and Sunday) on different routes of a road network and so the patterns of their trajectories change accordingly. We can broadly categorized seven days of a week into two groups: Group1 = {Mon, Tue, Wed, Thu, Fri} and Group2 = {Sat, Sun}. We define day distance (DD) in terms of a day difference between two natural days and the distance of the groups they belong to. Assume that a query trajectory Q has the start-day Qsd and the end-day Qed and a data trajectory T has the start-day Tsd and the end-day Ted.

Definition 7. If Q and T have motions on the same day, then day distance (Day-dist) between them equals to zero (0). Otherwise, it is equal to the day difference between them (considering cyclic order of days, ranging from 1 to 3).

For example, if Q has motion on Monday and T has motion on Friday, then Day-dist = 3 but not 4.

Definition 8. *If Q and T have motions on the same group of days, then group distance (Group-dist) between them equals to zero (0). Otherwise, Group-dist is greater than the maximum of Day-dist, i.e., five.*

Definition 9. If Q and T lie in different groups, then DD value between them is defined as

$$DD = Group\text{-}dist + Day\text{-}dist. \tag{6}$$

For example, if Q has motion on Wednesday and T has motion on Sunday, then DD = 5+3=8, where 5 is Group-dist and 3 is Day-dist.

Definition 10. Suppose that the motions of Q and T start on different days and end on different days. Trajectory day distance (TDD) between them can be given by the average of start Day-dist and end Day-dist.

$$TDD = (DD(Qsd, Tsd) + DD(Qed, Ted)) /2 \quad (7)$$

For example, the motion of Q starts on Monday and ends on Thursday while that of T starts on Tuesday and ends on Friday. TDD between Q and T is calculated as DD= (1+1)/2 = 1, where the former 1 is the distance between Monday and Tuesday and the later 1 is the distance between Thursday and Friday.

3.2.3 Temporal Similarity Based on Weeks (W)

To wide the scope of temporal similarity measure we select week distance (WD) rather than month or year distance because we can represent them in terms of weeks. For example, we can divide 5 years in terms of week-range as below:

1st year = 1st week (this week) up to past 52nd week
2nd year = previous 53rd - 104th weeks
3rd year = previous 105th - 156th weeks
4th year = previous 157th - 208th weeks
5th year = previous 209th - 260th weeks

Assume that a query trajectory Q has the start-week Qsw and the end-week Qew as well as a data trajectory T has the start-week Tsw and the end-week Tew.

Definition 11. Week distance (WD) can be defined as the week difference between two trajectories which is an absolute value.

Definition 12. Suppose that the motions of Q and T start from one week and end to another week. Trajectory week distance (TWD) can be given by the average of the start-week distance and the end -week distance between them.

$$TWD = (WD(Qsw, Tsw) + WD(Qew, Tew)) /2 \quad (8)$$

For example, if the motion of Q starts in the 1st week and ends in the 2nd week while that of T starts in the 5th week and ends in the 6th week, then TWD, between Q and T is calculated as TWD = (|1-5| + |2-6|)/2 = (4+4)/2 = 4, where the former 4 is the distance between the 1st week and the 5th week, and the latter 4 is the distance between the 2nd week and the 6th week.

3.2.4 Temporal Distance Measure

As mentioned earlier, temporal distance measure is expressed in terms of temporal distance (TD) which is a combination of three distances, namely TRD, TDD, and TWD.

Definition 13. Temporal distance (TD) between two trajectories, say Q and T, of moving objects on the road networks is defined as the summation of a distance triplet (TRD,TDD, TWD), where TRD, TDD and TWD stand for time range distance, day distance and week distance, respectively.

$$TD(Q, T) = \alpha*TRD + \beta*TDD + \gamma*TWD \quad (9)$$

where "α" , "β" and "γ" are temporal weights.

In addition, we get the near optimal values of "α", "β" and "γ" from our experiments. They are: "α" = 60, "β" = 20 and "γ" = 1.

By substituting the values of "α", "β" and "γ" in the above expression we get,

$$TD(Q, T) = 60*TRD + 20*TDD + TWD \quad (10)$$

3.2.5 Temporal Similarity Measure Algorithm

Temporal similarity measure algorithm computes the temporal distance between a given query trajectory and a data trajectory of moving objects with respect to their motions. Algorithm 2 shows a temporal similarity measure. In the first step of the measure, it computes the range distance (RD) of a trajectory Dt from the given Q by determining the range distance between their motions. In the second step, it computes the day distance between Dt and Q by determining the day difference between their motions. In the third step, the week distance (WD) between Q and Dt is computed by determining the week difference between their motions. In the final step, the temporal distance (TD) is computed by summing up all of the three temporal distances TRD, TDD and TWD.

Temporal Similarity Measure Algorithm:
Input: A query Q and a data trajectory (Dt)
Output: Temporal Distance (TD) of a trajectory with respect to Q

```
TemporalSimilarity(Q, Dt)
  {     temporalDistance=0;
        getTemporalDistance(Q, Dt, TRD, TDD, TWD);
        temporalDistance = α *TRD + β *TDD + γ* TWD;}
  getTemporalDistance(Q, Dt, RD, DD, WD)
  {     QR , QD , QW ;          //range, day and week distances of Q
        DtR, DtD, DtW;          //range, day and week distances of Dt
        TRD = 0, TDD = 0, TWD = 0;
        TRD = RD(QR, DtR); TDD = DD(QD, DtD);
        TWD = WD(QW, DtW); return;}
```

Algorithm 2. Temporal similarity measure

3.3 Spatio-temporal Similarity Measure

In this section, we define spatio-temporal similarity measure based on above mentioned two similarity measures, spatial measure and temporal measure. To express the spatio-temporal similarity for moving objects trajectories, we develop a spatio-temporal distance (STDist) function. The notion of the measure is: for any given spatio-temporal query Q, the most similar data trajectory is one which has the lowest STDist from the motion of the query Q.

3.3.1 Spatio-temporal Similarity

There are two approaches to define our spatio-temporal similarity measure. The first one is an addition-based approach (AA) and the second one is a multiplication-based approach (MA). Both approaches can be used depending on the situations of application types. Assume that there are two trajectories in a spatial network; one of them is a query trajectory Q and another one is a data trajectory T.

Definition 14. Spatio-temporal distance (STDist) between Q and T, using the addition-based approach (AA), can be defined as the average distance obtained by the spatial distance (SD) and temporal distance (TD) of T from the motion of Q.

$$STDist(Q, T) = (SD + \delta * TD) / 2 \tag{11}$$

where "δ" is the spatio-temporal weight and its good value is 20 by our multiple experiments. AA is useful to the situation where the complexity of road networks is low and easy to determine "δ".

Definition 15. Spatio-temporal distance (STDist) between Q and T, using the multiplication-based approach (MA), can be defined as the product of spatial distance (SD) and temporal distance (TD) of T from the motion of Q.

$$STDist(Q, T) = (SD / \delta) + 1) * (TD + 1) \tag{12}$$

"δ" has the same meaning as in AA. MA is useful to the situation where the complexity of road network is high and relatively difficult to determine δ.

3.3.2 Spatio-temporal Similarity Measure Algorithm

Spatio-temporal similarity measure algorithm computes spatio-temporal distance (STDist) between a given query trajectory Q and a data trajectory DT of moving objects with respect to their motions. Algorithm 3 is about spatio-temporal similarity measure. In the first step of the measure, it computes spatial distance (SD) of DT with respect to Q. In the second step, it computes temporal distance (TD) of DT by combining range distance, day distance and week distance. In the last step, it computes spatio-temporal distance (STDist) of DT by using its both distances SD and TD. Either addition-based approach or multiplication-based approach can be used to calculate STDist of DT with respect to the motion of Q depending on the situations of the road networks.

Spatio-temporal Similarity Measure Algorithm:
Input: A spatio-temporal query Q and a data trajectory (DT)
Output: Spatiotemporal Distance (STDist) of DT from Q
1. *Compute Spatial Distance (SD) // using spatial measure algorithm*
2. *Compute Temporal Distance(TD) // using temporal measure algorithm*
   $TD(Q, Dt) = \alpha * RD + \beta * DD + \gamma * WD$
3. *Compute Spatio-temporal Distance (STDist) using one of below two approaches:*
   (A) Multiplication Approach (MA) OR (B) Addition Approach (AA)

Algorithm 3. Spatio-temporal similarity measure

3.4 Trajectory Search Algorithm

In this section, we will describe our spatio-temporal similar trajectory search algorithm. To support it, we describe a segment finding algorithm based on signature files in the later part of the section.

3.4.1 Spatio-temporal Similar Trajectory Search Algorithm

For any given query trajectory, there must be an appropriate method to retrieve the most relevant data trajectories efficiently. To achieve this goal, we propose a new spatio-temporal similar trajectory search algorithm which can find a set of similar trajectories to the motion of a given query.

Spatio-temporal Similar Trajectory Search Algorithm:
Input: A spatio-temporal query trajectory Q and data trajectories
Output: Similar trajectories with Spatio-temporal Distance (STDist)
//Partial matching process based on common segment/s
1. *Using signature files, find all trajectories that match at least one segment of spatio-temporal query trajectory Q,*
   Result1= CandidateSet1;
2. *for all members of Result1, calculate SD and TD*
3. *for all members of Result1, compute Spatio-temporal Distance (STDist) using either of one approach:*
   *(A) STDist(Q, Dt) = (SD+ δ*TD)/2 // addition approach*
   (B) STDist(Q, Dt) =((SD/δ)+1)(TD+1) // multiplication approach*
4. *Sort the trajectories in the ascending order of STDist*
5. *Fetch k most similar trajectories (a user provides the value of k)*
   Result= k most similar number of trajectories

Algorithm 4. Spatio-temporal similar trajectory search

Algorithm 4 presents our spatio-temporal similar trajectory search method. The first step of the method selects CandidateSet1 for a given query Q by using signature files, where those trajectories are selected which match at least one of the segments of Q. The second step calculates both spatial distance (SD) and temporal distance (TD) of the candidate trajectories with respect to the given query. The third step computes the spatio-temporal distance (STDist) of the trajectories using either the addition-based approach or the multiplication-based approach depending on the complexities of networks. Then the fourth step sorts the trajectories based on the STDist of the trajectories in ascending order. So the trajectory with the lowest STDist takes the first position of the order indicating the most similar trajectory to the motion of a given query trajectory. The final step of the method fetches the K most similar trajectories to a given query Q where the value of k can be provided by a user.

3.4.2 Segment Finding Algorithm

To find a set of candidate trajectories in an efficient manner, we provide a segment finding algorithm. It can find a set of trajectories whose segment's signature covers at least one segment signature of a query trajectory. To do so, we modify and use the

method of Chang et al. [18] because they have already shown that their approach using signature files [19, 20] outperforms the popular access methods like FNR-tree and TB-tree in terms of retrieving trajectories. In their case, signature files are based on nodes of the road networks but in our case, they are based on the segments of the networks because it filters out irrelevant data trajectories while generating candidate sets and reduces the cost.

Segment Finding Algorithm:
Input: A query trajectory Q and data trajectories
Output: A priority queue of trajectories DT which match at least one segment signature of the query

```
DT = Φ ;
for Si segment in QT,                    // where    0< i <= n
 {        Sigi = grnerate_sig(Si );
          find_match_DT(Sigi, MDTi);
          DT = DT    MDTi ; }            // where i =1, 2, 3,……, n
```

Algorithm 5. Finding segment

Algorithm 5 presents our segment finding method. First, it generates signatures for all segments (Si) of a given query trajectory Q. Then, it searches those trajectories which match at least one of the segment signatures of Q. After that, candidate trajectories (MDTi) are stored into a priority queue DT, where a trajectory with the highest value of occurrence is stored first, a trajectory with the second highest value of occurrence is placed next and so on.

4 Performance Analysis

We implement our spatio-temporal similar measure algorithm under HP ML 150 G3 Server, Intel Zeon 3.0 GHz dual CPU with 2 GB main memory and 250GB HDD, using Windows 2003 Enterprise Server and Visual Studio.Net Compiler. Based on the road network of San Francisco Bay which consists of 170,000 nodes and 220,000 edges, we used the algorithm proposed by Brinkhoff [21] to generate 1000 trajectories of moving objects on the road network. We also generate 1000 query trajectories each with 40 segments on average. For retrieval time evaluation, we analyze the performances first by comparing our two spatio-temporal measures and then by comparing our both measures to that of Hwang's et al [17], which is the only existing work in the field of spatial networks to search similar trajectory considering real road network distance rather than Euclidean distance.

4.1 Addition-Based Approach Versus Multiplication-Based Approach

First, we compare our two spatio-temporal measures, addition-based approach (AA) and multiplication-based approach (MA), in terms of result coincidence i.e., the ratio

of the presence of the same trajectories in the result sets of the both approaches. Figure 1 shows a graph with the increasing number of k similar trajectories in x-axis and the ratio of the same trajectories that have been searched by the both methods in y-axis. It shows that with the increase value of k, the ratio of same trajectories is maintained high. For instances, the result coincidence is 99% in the case of two similar trajectories (k=2) and it is 92% where the value of k=5. Next, we compare similar trajectory retrieval time of our two approaches as shown in Figure 2. Both approaches have low retrieval time, i.e., about 1 second. This is because our trajectory search algorithm is based on signature files which enhance the searching speed.

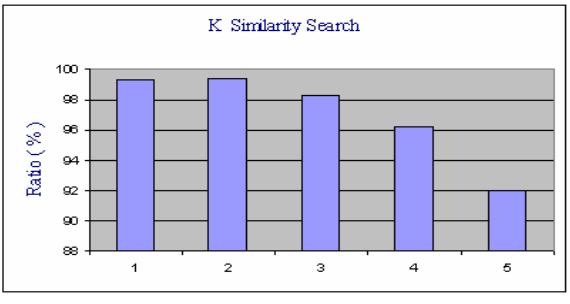


Fig. 1. Ratio of result coincidence between AA and MA

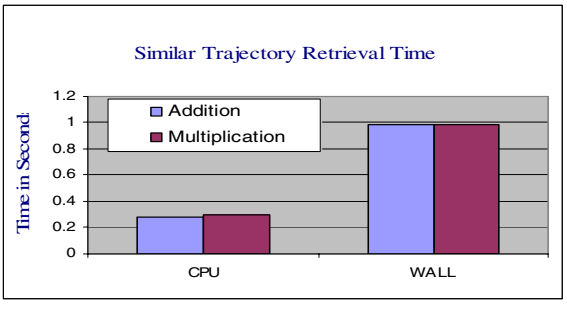


Fig. 2. Similar trajectory retrieval time for AA and MA

4.2 Our Methods Versus Existing Method

To show retrieval time efficiency, we compare our two methods, addition and multiplication, to the existing method proposed by Hwang et al [17]. As shown in Figure 3, in terms of similar trajectory retrieval time, our both methods are more than three-time faster than Hwang's method because our methods need just 0.2 second while Hwang's method needs 0.65 second to get the same result. So we can say that both of our methods better perform than that of the existing work of Hwang et al in terms of retrieval time.

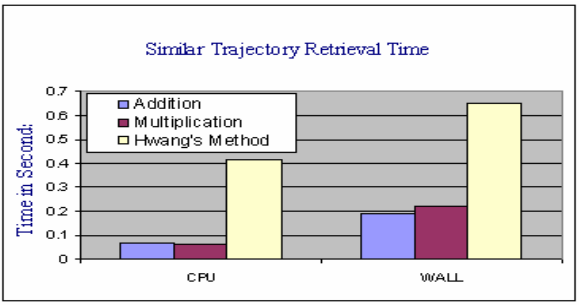


Fig. 3. Similar trajectories retrieval time comparison

5 Conclusions

Although there has been much research in the field of searching similar trajectories of moving objects in spatial databases, there exists less work to search similar trajectory on road networks based on spatio-temporal domains. In order to compute the similarity of moving object trajectories on road networks, we proposed three similarity measures: spatial measure, temporal measure and spatio-temporal measure. In addition, we designed a similar trajectory search algorithm to retrieve similar trajectories in an efficient manner by using signature files. For our performance analysis, we generate a data set by using a data generation algorithm made by T. Brinkhoff [21] based on the road network of San Francisco Bay and compared our methods to that of the existing work by Hwang et al [17]. Experimental results show that our approaches outperform the Hwang's work in terms of retrieval time for searching similar trajectories. As future work, we will prove the applicability of our approaches by doing some experiment on the real trajectories of moving objects on road networks.

Acknowledgments. This work is financially supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2005-211-D00 322). It is also supported by the Ministry of Education and Human Resources Development (MOE), the Ministry of Commerce, Industry and Energy (MOCIE) and the Ministry of Labor (MOLAB) though the fostering project of the Lab of Excellency.

References

1. Betke, M., Gips, J., Fleming, P.: The Camera Mouse: Visual Tracking of Body Features to Provide Computer Access for People with Severe Disabilities. IEEE Trans. Neural Syst. Rehabil. Eng. 10(1), 1–10 (2002)
2. Shimada, M., Uehara, K.: Discovery of Correlation from Multi-stream of Human Motion. Discovery Sci., 290–294 (2000)
3. Barbara, D.: Mobile Computing and Databases-A Survey. IEEE TKDE, 108–117 (1999)
4. Roddick, J.F., Hornsby, K.: An Updated Bibliography of Temporal, Spatial and Spatio-temporal Data Mining Research. In: Roddick, J.F., Hornsby, K. (eds.) TSDM 2000. LNCS (LNAI), vol. 2007, pp. 147–164. Springer, Heidelberg (2001)

5. Chudova, D., Gaffney, S., Mjolsness, E., Smyth, P.: Translation Invariant Mixture Models for Curve Clustering. In: Proc. of 9th SIGKDD, pp. 79–88 (2003)
6. Arikan, O., Forsyth, D.: Interactive Motion Generation from Examples. In: Proc. of ACM SIGGRAPH, pp. 483–490. ACM Press, New York (2002)
7. Valdes-Perez, R.E., Stone, C.A.: Systematic Detection of Subtle Spatio-temporal Patterns in Time-lapse Imaging: II. Particle Migrations. Bio-imaging 6(2), 71–78 (1998)
8. Pfoser, D., Jensen, C. S, Theodoridis, Y.: Novel Approaches in Query Processing for Moving Object Trajectories. In: Proc. of the 26th Intl. Conf. on VLDB, pp. 395–406 (2000)
9. Vlachos, M., Gunopulos, D., Kollios, G.: Robust Similarity Measures for Mobile Object Trajectories. In: Proc.of the 13th Intl. Workshop on DESA, pp. 721–728. IEEE Computer Society Press, Los Alamitos (2002)
10. Yanagisawa, Y., Akahani, J., Satoh, T.: Shape-Based Similarity Query for Trajectory of Mobile Objects. In: Proc.of the 4th Intl. Conf. on MDM, pp. 63–77 (2003)
11. Shim, C-B., Chang, J-W.: Similar Sub-Trajectory Retrieval for Moving Objects in Spatio-temporal Databases. In: Proc.of the 7th EECADIS, pp. 308–322 (2003)
12. Lin, B., Su, J.: Shapes Based Trajectory Queries for Moving Objects. GIS., 21–30 (2005)
13. Zeinalipour-Yazti, D., Song Lin, S., Gunopulos, D.: Distributed Spatio-Temporal Similarity Search. CIKM, 14-23 (2006)
14. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering Similar Multidimensional Trajectories. In: Proc.of the 18th ICDE, pp. 673–684. IEEE Computer Society Press, Los Alamitos (2002)
15. Sakurai, Y., Yoshikawa, M., Faloutsos, C.: FTW: Fast Similarity Search Under the Time Warping Distance. In: PODS, pp. 326–337 (2005)
16. Chen, L., Ozsu, M.T., Oria, V.: Robust and Fast Similarity Search for Moving Object Trajectories. In: ACM SIGMOD, pp. 491–502. ACM Press, New York (2005)
17. Hwang, J.-R., Kang, H.-Y., Li, K.-J.: Spatio-temporal Similarity Analysis Between Trajectories on Road Networks. In: ER (Workshops), pp. 280–289 (2005)
18. Chang, J.-W., Um, J.-H.: An Efficient Indexing Scheme for Moving Objects' Trajectories on Road Networks. In: Yu, J.X., Kitsuregawa, M., Leong, H.V. (eds.) WAIM 2006. LNCS, vol. 4016, pp. 13–25. Springer, Heidelberg (2006)
19. Faloutsos, C., Christodoulakis, S.: Signature Files: An Access Method for Documents and Its Analytical Performance Evaluation. ACM Tran. on Office Information Systems 2, 267–288 (1984)
20. Zobel, J., Moffat, A., Ramamohanarao, K.: Inverted Files Versus Signature File for Text Indexing. ACM Tran. on Database Systems 23, 453–490 (1998)
21. Brinkhoff, T.: A Framework for Generating Network-Based Moving Objects. GeoInformatica 6, 153–180 (2002)

Efficient Text Detection in Color Images by Eliminating Reflectance Component

Miyoung Choi and Hyungil Choi

Department of Media, Graduate school of Soongsil Univ., Korea
{utopialove, hic}@ssu.ac.kr

Abstract. This paper proposes a new approach to eliminate the reflectance component for the detection of text in color images. Color images, printed by color printing technology, normally have an illumination component as well as a reflectance component. It is well known that a reflectance component usually obstructs the task of detecting and recognizing objects like texts in the scene, since it blurs out an overall image. We have developed an approach that efficiently removes reflectance components while preserving illumination components. We decided whether an input image hits Normal or Polarized for determining the light environment, using the histogram which consisted of a red component. We were able to go ahead through the ability to extract by reducing the blur phenomenon of text by light because reflection component by an illumination change and removed it and extracted text. The experimental results have shown a superior performance even when an image has a complex background. Text detection and recognition performance is influenced by changing the illumination condition. Our method is robust to the images with different illumination conditions.

Keywords: reflectance component, text detection.

1 Introduction

Scene illumination is one of the most important factors that determine success or failure for many imaging and machine vision applications[1]. From the theory of the illumination-reflectance model, an image can be expressed in terms of its illumination and reflectance components. Due to the impossibility of controlling illumination conditions in real-world applications; image recognition, retrieval, segmentation and tracking, which are all invariant to illumination, emerge as challenging problems.

Most algorithms that the illumination variation must be controlled uniform or lighting. As for most existing studies, black-and-white scene text extraction with an image is provided in limited environment. We proposed a method which recognizes a change of illumination with a person recognizing and removing a reflection component by illumination in color images. We decided whether an input image hits normal or polarized for determining the light environment in a histogram. We used a homomorphic filter to remove the reflectance component. The homomorphic filtering is the technique that is based on the Fourier Transform, which enhances the contrast of image by reducing the low frequencies and amplifying the high frequencies in the frequency domain.

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1179 – 1186, 2007.
© Springer-Verlag Berlin Heidelberg 2007

2 System Overview

Generally, the extract of text in the image was applied to the image form of a black-and-white document with a little illumination or influence of light. However, the surface of the text domain containing color printing technology in the case of present can generate many problems in extracting a text or text region by light or a surface reflection by an illumination ingredient. The decrease text detection and recognition rate can arise from entirely blurred image influenced by the illumination or lose of edge information partially influenced by illumination. The image which can occur by influence of illumination is generally categorized as normal, polarized and backside or dark. However, this paper is defining the image which could occur by influence of illumination the form of normal and polarized light shown in Fig.1. Generally, we exclude the form of backside or dark, because it did not occur when illumination occurred from the back of a document.

(a) Normal(left), Polarized(right) (b) Normal(left), Polarized(right)

Fig. 1. Normal image and Polarized image

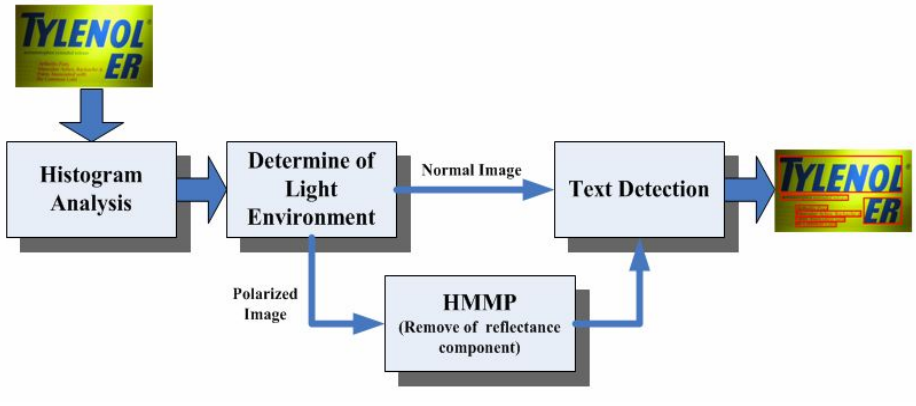


Fig. 2. Flow of the proposed algorithm

The approach is overviewed in Fig. 2. We distinguish a state of image by analysis of a histogram equal to a red ingredient of an input image. In the case of normal conditions the image detected the text region without handling an HMMP(Homomorphic

filter) process; we remove a coefficient ingredient equal to a reflectance ingredient by handling HMMP in the case of polarized.

3 Determination of Light Environment

To decide whether an input image hits normal or polarized for determining the light environment, we used a histogram which consisted of a red component in a color ingredient. The reason is because a red color ingredient is affected most by a state of illumination in three color ingredients. The following is the method that was suggested to determine the condition of an image.

In the first, we partitioned the S(Shadow), M(Midtone) and H(Highlight) domain about histogram of red component in a color image as shown in Fig. 3. We define a boundary that shadow domain is 0~79, midtone domain is 80~180 and highlight domain is 181~255.

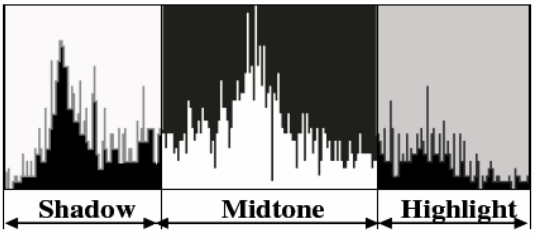
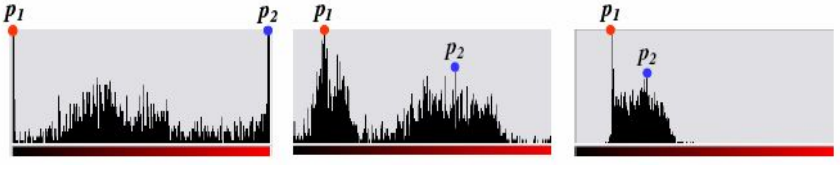


Fig. 3. Histogram of red component

The second one is that detect two pick in histogram as shown in Fig. 4. The method is detect two Pick using Ping-Sung Liao multi-threshold algorithm[2].



(a) Normal (b) Normal (c) Polarized

Fig. 4. Typical distributions of the red color histograms obtained under two lighting environments

The third one is that a distance threshold value between two pick is fixed and the significance is decided what kind of group two detected pick ingredients are included in among three groups(S, M, H). It shows in Fig.5 what histogram distribution of an image by illumination environment and a position of pick.

$$LightEnvironment = \begin{pmatrix} Normal \quad , \quad 70 \le d \le 130, & \begin{pmatrix} P_i, P_j \in S \\ or\ P_i, P_j \in H \end{pmatrix} \\ Polarized, \quad 140 \le d \le 210, & \begin{pmatrix} P_i \in S, P_j \in S \\ or\ P_i \in S, P_j \in M \\ or\ P_i \in M, P_j \in H \end{pmatrix} \end{pmatrix} \tag{1}$$

$$d = \left| P_i - P_j \right|$$

where P_i and P_j represent the two peak points in the histogram.

We accomplish text detection process without treating an HMMP filter if illumination environment of image does normal. If illumination environment of image is polarized, we take coefficient of a Reflectance ingredient with an HMMP filter and remove it, after accomplish text detection process.

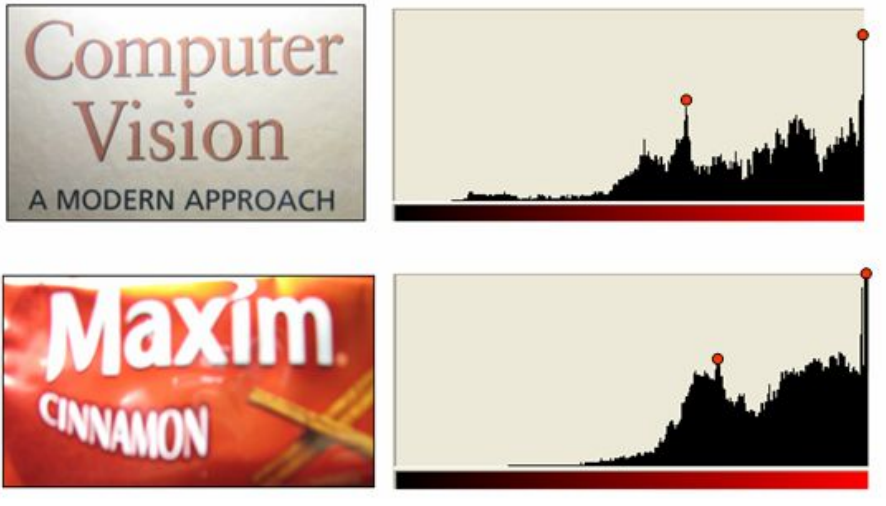


Fig. 5. Histogram distribution and position of picks

4 Eliminating Reflectance Component Based on Homomorphic Filtering

Homomorphic Filtering is the technique that is based on Fourier Transform, which enhances the contrast of image by removing the low frequencies and amplifying the high frequencies in frequency domain[3-5].

To eliminate the uneven scene illumination, a homomorphic filter was used. In many realistic situations the illumination component varies slowly, while the reflectance component changes rapidly. The homomorphic filter, given in Eq(3) if applied

in the frequency domain, as shown in Fig 6. It is assumed that an input image is the multiplication of illumination and reflectance components, and the contrast in the image can be emphasized by adjusting D_0 in Eq(3).

$$\ln[F(x,y)] = \ln[I(x,y)] + \ln[R(x,y)] \tag{2}$$

$$H(u,v) = (\gamma_H - \gamma_L)[1 - e^{-c(D^2(u,v)/D_0^2)}] + \gamma_L \tag{3}$$

This procedure should decrease the effect of non-uniform illumination in the image and enhance the details of the scene. Illumination tends to vary slowly, or relatively, across an image[6]. On the other hand, reflectance is characterized by sharp changes, especially at boundaries. Illumination corresponds to low frequency and reflectance corresponds to high frequency in frequency domain.

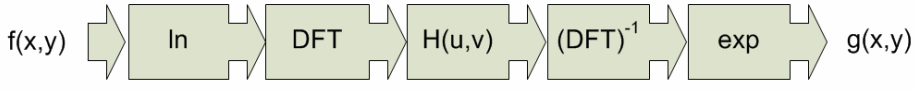


Fig. 6. Homomorphic filtering approach

The removed Reflectance ingredient was the same by an experiment. We considered a price of 0.75~1.25 in a Reflectance ingredient and took it and removed it. Fig 7 is image that removed Reflectance component.

Fig. 7. This image removed reflectance component

5 Detection of Text in Color Image

In this section is composed of two phases, the first phase is global optimal thresholding method[3][7] that segment foreground from the background, the second phase is run length smoothing algorithm(RLS)[8][9] that detect the text from extracted foreground.

Low resolution image contrast makes it difficult to segment foreground from the background, the corresponding histogram peaks will tend to overlap. We use a method for estimating thresholds that produce the minimum average segmentation error.

If the variances are equal, $\sigma^2 = \sigma_1^2 = \sigma_2^2$, a single threshold is sufficient:

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 + \mu_2} \ln\left(\frac{P_2}{P_1}\right) \tag{4}$$

where μ_1 and σ_1^2 are the mean and variance of the Gaussian density of one class of pixel(object) and μ_2 and σ_2^2 are the mean and variance of the other class(background).

If $P_1 = P_2$, the optimal threshold then T is the mid-value of the mean values: $T = \frac{\mu_1 + \mu_2}{2}$. Every point in the image had been assigned a threshold. Finally, a binary decision was carried out for each pixel using the rule:

$$f(x, y) = \begin{cases} 1 & if\ f(x, y) \geq T \\ 0 & otherwise \end{cases} \tag{5}$$

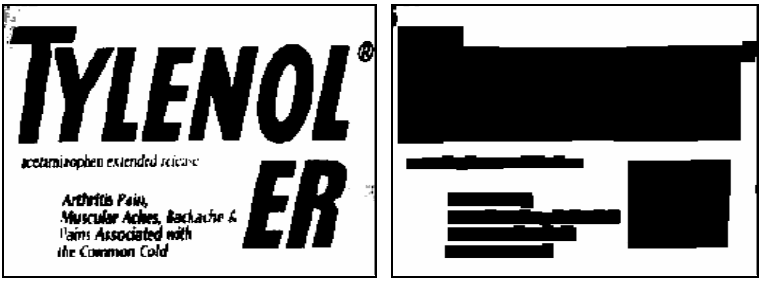


Fig. 8. The result of applying the global optimal thresholding(left) and the run length smoothing algorithm(right)

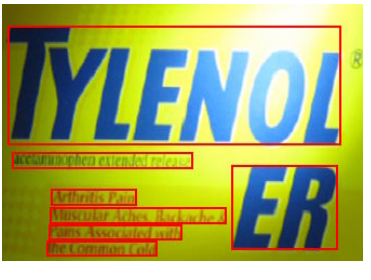


Fig. 9. Detected text

This phase is RLS is usually followed by connected component analysis and used for image segmentation. RLS is based on examining the white runs existing in the horizontal and vertical directions. For each direction, implemented by applying for horizontal smoothing(a left-right) or for vertical smoothing(top-down) scan of the image and then by selecting all whit runs with length less than a given threshold. The

final image is extracted after applying and AND operation at the horizontal and vertical smoothed images. Connected components are detected in the RLS image. We calculate statistics of the regions using connected components and extract the text.

6 Experiments and Discussion

The experiments, the proposed method remove a reflection component by illumination, which recognize a change of illumination with a person recognizing, in color

Table 1. OCR recognition rate with and without the incorporation of the proposed method

	Normal images	Polarized images
FineReader6 or Aremi6.5	72.1 %	61.3 %
Proposed method + FineReader6 or Aremi6.5	-	70.4 %

(a) (b) (c) (d)

Fig. 10. Experiment result images (a) Result image (b) Gray image (c) Removed reflectance component of image (d) Binary image

images based on the explosion of digital cameras or camera phones. The proposed algorithm has been implemented using Visual C^{++}. Our experiment environment is Intel Pentium 4 CPU 3.0GHz, 1GB RAM, experimented on around 500 pieces color images that normal image is 150 pieces and polarized image is 350 pieces. We decided whether an input image hits normal or polarized for determine of light environment, use the histogram which consisted of a red component. We could able to go ahead through extractability ability by reducing a blur phenomenon of text by light because reflection component by an illumination change and removed it and extracted text. And global optimal thresholding method is effective for extracting text from noise background. We compared the results obtained by the well-known OCR engine ABBYY FineReader_6[10] and Aremi_6.5[11] with and without the incorporation of the proposed technique. As shown in the representative results at Table 1, the application of the proposed method has shown best performance.

Acknowledgment

This research is supported by Seoul R&BD Program(10581cooperateOrg93112).

References

1. Almas Anjum, M.: A New Approach to Face Recognition Using Dual Dimension Reduction. IJSP 2(1) (2005)
2. Liao, P.-S., Chen, T.-S., Chung, P.-C.: A Fast Algorithm for Multilevel Thresholding. Information Science and Engineering 17 (2001)
3. Jain, A.: Fundamentals of Digital Image Processing. Prentice-Hall, Englewood Cliffs (1989)
4. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice-Hall, Englewood Cliffs (2002)
5. Myler, H.R., Weeks, A.R.: The Pocket Handbook of Image Processing Algorithms in C. Prentice Hall, Englewood Cliffs (1993)
6. Parker, J.R.: Algorithms for Image Processing and Computer Vision. Wiley computer publishing, Chichester (1997)
7. Umbaugh, S.E.: Computer Imaging: Digital Image Analysis and Processing. CRC Press, Florida (2005)
8. Papamarkos, N., Tzortzakis, J., Gatos, B.: Determination of run-length smoothing values for document segmentation, ICECS (1996)
9. Liu, Q.H.: Newspaper Headlines Extraction from Microfilm Images. IEEE, Los Alamitos (2002)
10. ABBYY.: (2005) www.finereader.com
11. Aremi.: (2006), http://www.perceptcom.com

Enhanced Non-disjoint Multi-path Source Routing Protocol for Wireless Ad-Hoc Networks*

Moon Jeong Kim[1], Dong Hoon Lee[1], and Young Ik Eom[2]

[1] Graduate School of Information Management & Security,
Center for Information Security Technologies (CIST), Korea University,
Anam-dong Seongbuk-Gu, Seoul, 136-701, Korea
[2] School of Information and Communication Engineering,
Sungkyunkwan University,
300 Cheoncheon-dong Suwon, Gyeonggi-do 440-746, Korea
{tops, donghlee}@korea.ac.kr, yieom@ece.skku.ac.kr

Abstract. A wireless ad-hoc network is a temporal network formed by a collection of wireless mobile nodes without the aid of any existing network infrastructure or centralized administration. This network is suitable for a sensor network and for ubiquitous computing environments. In this paper, we suggest a node/link non-disjoint multi-path source routing protocol for a wireless ad-hoc network. This protocol also provides Internet service to mobile hosts in a wireless ad-hoc network. Those mobile hosts have no overhead as a result of the protocol because base stations take the responsibility for connectivity. Even though a route inevitably fails during a specific service, our proposed scheme reduces the overhead of route request flooding and of route re-establishments. The proposed scheme re-registers by maintaining node/link non-disjoint multi-paths between a mobile node and a base station.

Keywords: enhanced wireless ad-hoc network, multi-path routing protocol, node/link non-disjoint route, on-demand routing protocol.

1 Introduction

In wireless ad-hoc network environments, little or no network infrastructure may be available for use or the use of an available network infrastructure may be undesirable because of cost or convenience. Some examples of possible uses of a wireless ad-hoc network include military applications, law enforcement, emergency response efforts, commercial use, and education[1],[2]. Routing protocols have been the subject of intensive research in the wireless ad-hoc network area. In recent years, many multi-path ad-hoc routing protocols that can operate more quickly and adaptively even when a route failure occurs[3],[4],[5] have been

* "This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement)" (IITA-2006-(C1090-0603-0025)).

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 1187–1196, 2007.
© Springer-Verlag Berlin Heidelberg 2007

developed. Particular interest is given to the global Internet connectivity for wireless ad-hoc networks[6],[7],[8].

In this paper, we propose a simple and efficient on-demand routing protocol that can establish and maintain link/node non-disjoint multiple paths in a wireless ad-hoc network. Our protocol provides Internet service to mobile hosts in a wireless ad-hoc network. Our protocol can reduce the time for the route recovery process, the control packet overhead, and the end-to-end delay from the source to destination. As a consequence with our protocol, faster route re-establishment is possible. The rest of the paper is organized as follows. In Sections 2 and 3, we present two routing protocols that maintain multi-paths for a wireless ad-hoc network. We give briefly an introduction of other extension approaches, respectively. In Section 4, we give a basic operation and some algorithms for the proposed scheme. In Section 5, we present the simulation results. Finally, in Section 6, we present the conclusion.

2 A Multi-path Routing Protocol

2.1 The Scheme Using Disjoint Routes

A multi-path extension for the DSR protocol is proposed for quick re-establishment of new routes in the case of route failure[2],[3]. The extension explores two kinds of paths with the consideration of alternate disjoint routes as shown in Figure 1. When the primary route breaks, the shortest remaining alternate route is used.

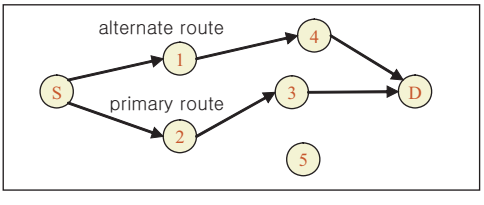


Fig. 1. The effect of the disjoint route

The key advantage of the protocol is the reduction in the frequency of route discovery flood. However alternate routes, in practice, always tend to be longer than the primary routes. This is especially true when we focus only on disjoint routes. Also, too many routes may be maintained because there is no limit on the number of paths that can be maintained in each mobile node.

2.2 A Scheme Using Non-disjoint Routes

We designed an on-demand, node/link non-disjoint, and multi-path routing protocol for wireless ad-hoc networks. This effect of the non-disjoint routes is shown in Figure 2.

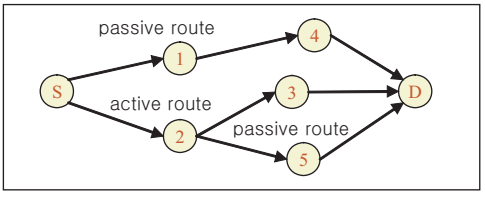


Fig. 2. Teh effect of the non-disjoint route

In this protocol, the number of paths from a source to a destination may be appropriately limited. A node receives a route request packet then determines whether it is duplicated or not. If it is not duplicated then the node forwards the packet. So this protocol is better by far in flooding overhead compared with the node/link disjoint multi-path routing protocol.

3 Extension Approaches

Two classes of approaches have been proposed to support extension of a wireless ad-hoc network to the Internet[6],[7],[8]. Proactive approaches flood advertisements from nodes acting as Mobile IP foreign agents through the whole wireless ad-hoc network. Reactive approaches allow the mobile nodes to broadcast solicitations to find foreign agents as they are needed. Typically, reactive approaches are more suitable for a wireless ad-hoc network environment than proactive approaches are. This is because of the nature of wireless communication. But, these reactive approaches cause a data transmission delay and the control packet overhead of a route re-establishment and re-register when a route fails by maintaining a single path between a mobile host and a Mobile IP foreign agent.

4 Enhanced Non-disjoint Multi-path Source Routing

We have published non-disjoint multi-path source routing protocol for a wireless ad-hoc network. The non-disjoint scheme has lower overhead than the disjoint scheme because intermediate node does not check and rebroadcast all route request packets received[9]. The protocol establishes non-disjoint multi-path between a source host and a GA. Also we have published the transformation scheme between our non-disjoint routing protocol and Mobile-IP[10].

4.1 Terminology

In this section, we define the following terminology:

GA (Gateway Agent)**:** a base station that supports both our protocol and a Mobile IP, as well as an interconnection scheme between them.

Active Route: a path on which a source host is sending data packets to a destination host (or a GA).
Passive Route: a path that may be used when an active route fails.
RQP (Route reQuest Packet)**:** a control packet that a source host broadcasts in order to find a destination host (or a GA). When a source host wants to find a GA, the destination address of the RQP must be set to the IP limited broadcast address (255.255.255.255).
RRP (Route Reply Packet)**:** a control packet that the destination host received a RQP sends to it including the source route to the source host.
REP (Route Error Packet)**:** a control packet that the node discovers a link failure over an active or passive route and then notifies this fact to the source or destination host.
MQP (Member reQuest Packet)**:** a control packet that the source host transmits to an appropriate GA in order to send a registration request.
MRP (Member Reply Packet)**:** a control packet that the GA receives a registration reply message and then sends it to the source host.
RC: a data structure containing multi-path information needed by the node.
VLE (Visitor List Extension)**:** a data structure that is maintained to manage mobile hosts in wireless ad-hoc networks by each GA.

4.2 The Basic Operation

The protocol is able to communicate between nodes within a wireless ad hoc network and support nodes that want Internet service. Our protocol is composed of a GA Discovery Phase, a Member Registration Phase, and a Service Delivery Phase. GA Discovery is the phase by which a node S wishing Internet service obtains one or more source routes to the GA. Member Registration is the phase by which the node S already knows a route to an appropriate GA and so registers itself to its HA(Home Agent) through the GA. Service Delivery is the phase by which the node S is provided Internet service through its GA.

GA Discovery Phase. When a node in a wireless ad-hoc network wants Internet service, first it initiates the GA discovery phase as shown in Figure 3. In Figure 3, when a source host(S) finds GAs(GA1, GA2), it then selects an appropriate GA. This source then registers itself to its HA through the selected GA. During a route request, each node receives a RQP and then rebroadcasts it if it has not been previous received. Each GA receives a RQP and shall reply for a limited multi-path.

Each host, which is included in the paths maintains the route information in its RC and monitors the link status of its neighboring nodes on the path. Whenever a link is disconnected, the node informs this fact in the opposite direction of the route in its RC. In this way, each node on the route maintains its RC that includes the latest-fashioned information.

Our protocol has advantages from the view-point of route re-establishment cost and average end-to-end data transmission delay. Also it does not have a

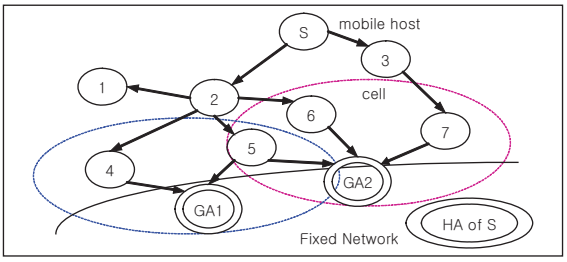


Fig. 3. The GA Discovery Phase

high overhead because of the non-disjoint route. We assume that each participant node has a wireless interface that is able to sense the link status of its neighboring nodes. We also assume that all the nodes in the network have routing facilities.

Member Registration Phase. On complete GA Discovery Phase, the S may register its HA through selected GA as shown in Figure 4.

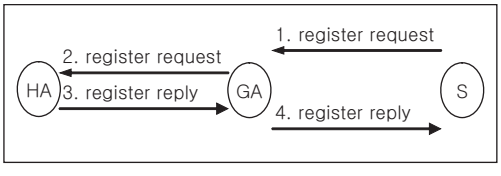


Fig. 4. The Member Registration Phase

In Figure 4, the GA receives a MQP from S and translates this MQP to a registration request message for the Mobile-IP. It then transmits it to the HA of S. On receiving the registration reply packet from the HA of S, the GA converts it into a MRP and then transmits it to S. On completing the Member Registration Phase, the S may be delivered Internet services. This processing is called the Service Delivery Phase.

4.3 Algorithms

In this section, we illustrate the detailed procedure of the proposed protocol.

GA Discovery Phase. For Internet service, a source host in a wireless ad-hoc network first searches its RC for the route to any GA. In the case that the source host has information on the route to a GA in its RC, it can transmit a MQP to the GA along the route. Otherwise, it performs the task shown in Table 2.

An intermediate node receives a RQP, determinates whether it is a duplicated, appends its address to end of it, and rebroadcasts it. An intermediate node receives a RRP and forwards it to the next host on the path in the RRP. A source host transmits a MQP to the GA by using the route information in the first RRP.

Table 1. Alg. for GA discovery

```
Tw: timeout value
{
  create a RQP and broadcast it;
  wait(Tw);
  while (not having received a RRP during Tw) {
    update the Tw and TTL;
    broadcast a new RQP;
    wait(Tw);
  }
}
```

Member Registration Phase. The procedure for the member registration request of the source host is showed in Table 2.

Table 2. Alg. for S in the Member Registration Phase

```
Tw: timeout for waiting MRP
{
  create a MQP and transmit it to the GA of the source host;
  wait(Tw);
  while (not having received a MRP during Tw) {
    perform a GA Discovery Phase;
  }
}
```

If a source host does not have any MRP during a given timeout, it repeats the GA Discovery Phase and Member Registration Phase. In this case, if it has some

Table 3. Alg. for the GA in the Member Registration Phase

```
RQM: Registration reQuest Message
RRM: Registration Reply Message
{
  if (Lifetime great than zero)
    insert a new entry in the VLE;
  else
    delete the entry in the VLE;
  transform the MQP to RQM;
  send the RQM to the HA of the source host;
  wait(received RRM);
  update the entry in the VLE;
  transform the RRM to MRP;
  send the MRP to the source host;
}
```

passive route(s), it may achieve the immediately Member Registration Phase again. An intermediate node having received a MQP forwards it to the next host on the path in the MQP. Each GA manages mobile hosts in wireless ad-hoc networks by maintaining a VLE. The procedure for the Member Registration Phase of the GA is showed in Table 3.

When a source host wants to stop Internet service, it transmits a MQP with the `Lifetime` field set to a value of zero in order to achieve a registration release. An intermediate node receives a MRP and forwards it to next host on the path in the MRP.

5 Performance Evaluations

To simulate the proposed protocol, we used the simulation tool called Simlib[11]. We describe some assumptions and simulation environments in Section 5.1 and present the analysis of the simulation results in Section 5.2.

5.1 Simulation Environments

We assumed that each node had a radio transmission range of 250m and that the transmission delay between neighbor nodes was 30ms. Each run was executed for 1180seconds of simulation time and we measured the performance during the interval [1000s, 1180s]. We assumed that the data packet was 512 CBR (Constant Bit Rate) and 4 data packets per second were generated. Our simulation model was based on the simulation model of [12],[13]. In our simulation, a random waypoint model was used as a mobility model, and the pause time was set to zero[14].

5.2 Performance Analysis

We performed the simulation for three cases such that the number of paths maintained in each node was 1, 3, and 5 as shown in Figure 5. The throughput was obtained by dividing the number of data packets correctly received from a GA by the simulation time. The end-to-end delay meant the time elapsed for a packet to travel from its source to the GA. Our simulation modeled a network of 40 mobile nodes placed randomly within a 700m x 500m closed area.

Figure 5-(a) presents the throughput for the three cases according to different average movement speeds of the mobile nodes. The difference of throughputs between using three paths and using a single path with average movement speed of 10m/s was approximately 320% larger than the difference of throughputs between using three paths and using single path at an average movement speed of 2m/s. In Figure 5-(b), we can see that single path based routing had much longer end-to-end delay when the average was over 6m/s. These were because, as mobility increased, more route reconstructions were necessary and more data packets should have stayed in the source nodes during the longer route recovery process.

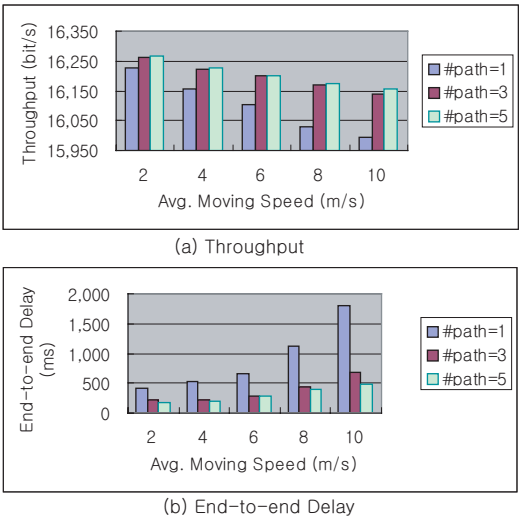


Fig. 5. Single Path vs. Multiple Path

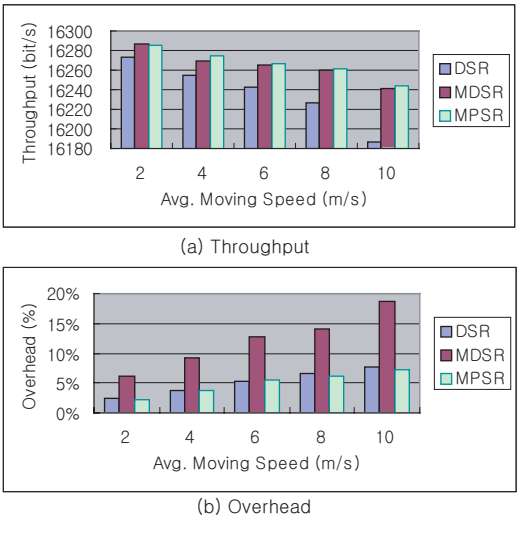


Fig. 6. Disjoint Route vs. Non-disjoint Route

In addition, we compared the throughput and overhead of our protocol maintaining three non-disjoint routes, called by the MPSR, with the DSR protocol and MDSR protocol using disjoint routes. The simulation results are shown in Figure 6. The overhead was obtained by dividing the number of transmitted control packets by the number of data packets that correctly reached to the GA.

Our simulation modeled a network of 10 mobile nodes placed randomly within a 500m x 300m closed area.

In Figure 6-(a), we can see that the two multi-path protocols, MPSR and MDSR, were more stable than the DSR. But in Figure 6-(b), the MPSR and DSR protocols had similar overhead, as the mobility increased. This is because the DSR protocol incurred more control packets for route re-discovery and re-establishment; however, the MPSR protocol incurred more control packets for route maintenance. The MDSR protocol had higher overhead than the others because an intermediate node did not check and rebroadcast all route request packets received.

6 Conclusion

In this paper, we suggested a node/link non-disjoint multi-path source routing protocol for a wireless ad-hoc network. This protocol also provided Internet service to mobile hosts in a wireless ad-hoc network. To do so, the GA supported both the protocol for a wireless ad-hoc network and the Mobile-IP, had multi-paths with the source host in a wireless ad-hoc network, and acted as a gateway in order to provide Internet service. This protocol put the BS in charge of function that translated from (to) the packet for the wireless ad-hoc network into (from) the packet for Mobile-IP. As a consequence the overhead of the mobile host was free. To compare with existing interconnections, we simulated cases using a single path and multi-paths. We compared our scheme with the scheme of node/link disjoint route multi-path.

With the great interest now in home networks, sensor networks, pervasive computing and ubiquitous computing applications, research on the expansion of as well as research on routing protocol in a wireless ad-hoc network has been the subject of intensive efforts. Our scheme contributes a foundation skill to enhance a wireless ad-hoc network to other technologies such as ubiquitous computing.

References

1. Corson, S., Macker, J.: Mobile ad hoc (MANET): Routing Protocol Performance Issues and Evaluation Considerations, RFC 2501, IETF (1999)
2. Johnson, D.B., Maltz, D.A., Hu, Y.: The Dynamic Source Routing Protocol for Mobile Ad-hoc Networks (DSR), Internet Draft, IETF (2004)
3. Nasipuri, A., Das, S.R.: On-Demand Multipath Routing for Mobile Ad Hoc Networks. In: Proc. of International Conf. on Computer Communications and Networks, pp. 64–70 (1999)
4. Valera, A., Seah, W.K.G., Rao, S.V.: Cooperative Packet Caching and Shortest Multipath Routing in Mobile Ad Hoc Networks. In: Proc. of the 22th Annual Joint Conf. of the IEEE Computer and Communications Societies(INFOCOM), vol. 1, pp. 260–269 (2003)
5. Hwang, I., Yeh, C., Wang, C.: Link Stability, Loading Balance and Power Control Based Multi-path Routing (SBPMR) Algorithm in Ad Hoc Wireless Networks. In: Proc. 10th International Conf. on Telecommunications, vol. 1, pp. 406–413 (2003)

6. Ratanchandani, P., Kravets, R.: A Hybrid Approach to Internet Connectivity for Mobile Ad Hoc Networks. In: Proc. IEEE International Conf. on Wireless Communications and Networking, pp. 1522–1527 (2003)
7. Sun, Y., Belding-Royer, E.M., Perkins, C.E.: Internet Connectivity for Ad Hoc Mobile Networks. International Journal of Wireless Information Networks special issue on Mobile Ad Hoc Networks(MANETs): Standards, Research, Applications, 1–15 (2002)
8. Nordstrom, E., Guningberg, P., Tschudin, C.: Gateway Forwarding Strategies for Ad Hoc Networks. In: 4th Scandinavian Workshop on Wireless Ad Hoc Networks (2004)
9. Kim, M.J., Kim, B.H., Lee, D.H., Eom, Y.I.: Non-Disjoint Multi-Path Source Routing Protocol for Ubiquitous Network. In: 1st Internatonal Conference on Ubiquitous Information Technology & Applications, pp. 424–433 (2007)
10. Kim, M.J., Lee, D.H., Eom, Y.I.: A Transformation Scheme for Enhanced Wireless Ad-hoc Networks. In: International Conference on Communications in Computing, pp. 36–42 (2007)
11. Law, A.M., Kelton, W.D.: Simulation Modeling and Analysis, 3rd edn. McGraw-Hill, New York (2000)
12. Boukerche, A.: A Simulation based Study of On-demand Routing Protocols for Ad Hoc Wireless Networks. In: Proc. 34th Annual on Simulation Symposium, pp. 85–92 (2001)
13. Clausen, T.H., Jacquet, P., Viennot, L.: Comparative Study of Routing Protocols for Mobile Ad-hoc Networks. In: Proc. International Conf. on Parallel Processing Workshops, pp. 124–137 (2002)
14. Camp, T., Boleng, J., Davies, V.: A Survey of Mobility Models for Ad Hoc Network Research, Wireless Communications and Mobile Computing (WCMC). Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications 2(5), 483–502 (2002)

Author Index

Abbas, Cláudia J. Barenco II-489
Abbas, Zulkifly II-13
Abdul-Rahman, Alias I-151
Abdullah, Azizol III-763
Abdullah, M. I-748
Abdullah, Salwani III-611
Abellanas, Manuel I-1
Aguilar R., Manuel III-575
Ahiska, S. Sebnem I-779
Ahmad, Fatimah II-13
Ahn, Hyung-Jin III-920
Ahn, Sangim II-296
Ahn, Suhong II-907
Ahn, Sukyoung II-226
Ahn, Sungyong I-252
Al-Hamadi, Ayoub I-397
Ali, Hasimah II-85
Alimohammadi, Abbas II-308
Aloulou, Mohamed Ali III-1027
Alptekin, S. Emre I-832
Álvarez, Manuel II-322
Amintoosi, M. III-495
Amiri, A. III-495
Appadu, A.R. III-774
Artigues, Christian III-1027
Aslan, Burak Galip I-1133
Aujla, Ramanpreet Kaur III-749
Ayob, Masri III-611

Babaei, Hamideh II-804
Badea, Bogdan III-460
Baeg, Sung Ho I-346
Baek, Kikyung III-858
Baek, Myung-Sun III-950
Baek, SeongEun I-1024
Bahn, Hyokyung I-201, I-252
Bajuelos, Antonio I-1
Balint-Kurti, Gabriel G. III-1114
Balouki, Youssef III-45
Barbatti, Mario I-281
Barsky, Brian A. III-1124
Bärwolff, Günter III-475
Basave T., Rosy III-674
Bei, Jia I-315
Bellas, Fernando II-322
Benkner, Siegfried I-281
Bezerianos, Anastasios III-566
Bhak, Jonghwa II-639
Bista, Rabindra III-1165
Blasco, G. II-727
Boada, Imma II-727, II-861
Bogdanov, Alexander III-1114
Bougioukos, Panagiotis III-555, III-566
Bouhdadi, Mohamed III-45
Brandic, Ivona I-281
Bui, Ngot Phu III-140

Caballero-Gil, P. III-544
Cacheda, Fidel II-322
Cao, Zhen II-514
Carbonell, Mildrey II-540, II-549
Carneiro, Víctor II-322
Carrasco, Alejandro III-811
Castilla V., Guadalupe III-575
Cattani, Carlo I-490
Cavouras, Dionisis III-239, III-555, III-566
Cedrés, David B. I-360
Cha, Byungrae III-201
Cha, Jaehyuk I-224, I-1146
Cha, JeongHee I-1083
Chabbar, El maati III-45
Chae, Junghwa I-500, III-929
Chai, Kevin III-724
Chang, Byoungchol I-1146
Chang, Elizabeth II-346, III-227, III-724
Chang, Jae-Woo III-1165
Chee, Vee Liem II-975
Chen, Hanxiong III-821
Chen, Hsiao Ching III-625
Chen, Jiming I-315
Chen, Yangzhou I-480, III-69
Chen, Yen Hung III-520
Chen, Yumin III-1152
Cheng, Hongju II-247
Cheon, Young Min I-1003
Cheong, Jae Youn III-585

Chiong, Raymond III-683
Chiu, Chuang-Cheng I-1107
Cho, Dongsub II-996
Cho, Hyun-Sook II-1097
Cho, Seokhyang II-996
Cho, Sung Won III-585
Cho, Tae Ho II-573, III-637
Cho, Yong Ju III-11, III-20, III-33
Cho, Yongyun I-335, II-818, II-829
Cho, Youngsong II-639
Choi, Hyung-Il I-981, I-1003, I-1014, I-1045, I-1074, III-1179
Choi, Inhwa I-1096
Choi, Jaeyoung I-325, I-335, II-829, II-839
Choi, Ji-hyeon II-131
Choi, Jongmyung II-849, III-739
Choi, Jongsun I-335, II-829
Choi, Ki-Moon II-455
Choi, Kwang-Soon II-593
Choi, Kyong Ho II-185
Choi, Miyoung III-1179
Choi, ShinHo I-164
Choi, SuGil I-912
Choi, Sun II-131
Choi, Sung-il III-486
Choi, Sungsoo I-1096
Choi, Yong S. I-1146
Choi, Young II-706
Choi, Young-Geun II-1085, III-886
Choi, YoungSik II-397
Chong, Kiwon II-296
Choo, Hyunseung II-216, II-275, II-286, II-469, II-1072, III-534
Chuah, Hean-Teik I-951
Chung, Chin Hyun III-790
Chung, I-Ping I-860
Chung, Ki-Dong I-689
Chung, Min Young II-216
Chung, Myoung-Bum I-961
Chung, Shu-Hsing I-767, I-860
Chung, So Young III-1
Chung, TaeChoong III-140
Chung, Tai-Myoung II-195, II-1107
Chung, Yongwha III-1141
Cobo, Angel II-680
Cordero, Rogelio Limon III-104
Cracknell, Arthur P. I-1054, III-410
Cruz R., Laura III-575, III-674
Cruz-Chávez, Marco Antonio III-697
Cui, Pingyuan I-480
Cui, Younggang II-987

d'Anjou, Alicia III-798
Dai, H.K. I-937
Dai, Liangtie III-81
Daman, Daut III-128
Darus, Maslina I-385
Daskalakis, Antonis III-239, III-555, III-566
Datta, Amitava II-627
Dauhoo, M.Z. III-774
Deinega, A. III-213
Deng, Hui II-514
Deraman, Aziz II-448
Deris, M. Mat III-115
Dévai, Frank II-51
Diez, Yago I-26
Dillon, Tharam II-346
Din, Der-Rong I-846
Du, YaJun II-434
Dumitru, O. III-445

Echegoyen, Zelmar III-798
Edwards, Doug III-154
Encheva, Sylvia I-305
Eom, Jung-Ho II-195
Eom, Young Ik III-1187
Ercan, M. Fikret II-1, III-309
Espinola, Jesús II-680

Fabregat, Ramon II-861
Fadzlah, Amalina Farhi Ahmad II-448
Farkas, Zoltan III-872
Fathi, Mahmood III-367, III-495, II-804
Feixas, Miquel II-602, II-727
Fernández-Medina, Eduardo III-262
Ferrer, Laia I-791
Fraire H., Héctor III-575, III-674
Frisch, Michael J. I-265
Fung, Yu-Fai III-309
Furuse, Kazutaka III-821
Fúster-Sabater, Amparo III-544, III-597

Gálvez, Akemi II-680
Gansterer, Wilfried I-281
García, Alberto I-791
García, I. III-941
Garzón, E.M. III-941

Gavrilova, Marina L. I-136, II-51
Gelbukh, Alexander I-424
Geng, Zhi I-678
Georgiadis, Pantelis III-239, III-555
Gervasi, Osvaldo I-281
Gevorkyan, Ashot S. III-1114
Ghodsi, Mohammad I-68
Ghourabi, Fadoua II-653
Goi, Bok-Min I-644, I-951
Gomes, Abel II-666
Gómez, Ariel III-811
Gomez, Antonio F. II-1038
Gómez S., Claudia G. III-575
Gong, Jianya III-1152
Gonzılez B., Juan J. III-575
Gorin, Andrey III-507
Graña, Manuel III-798
Greene, Eugene I-41
Guo, Hua I-82
Guo, Jianping III-1106
Guo, Wanwu III-81
Guo, Yufei II-559
Gutiérrez, Carlos III-262

Ha, Hojin I-622
Ha, JeungYo I-981
Hai, Tran Hoang II-383
Ham, Dall-ho I-1146
Hamdan, Abdul Razak III-611
Han, In Kyu II-157
Han, JinHee I-912
Han, Jung-Soo III-978, III-987, III-997
Han, Sang-Wook I-710
Han, Seokhee I-238
Han, Song III-227
Han, Young-Ju II-195, II-1107
Hani, Ahmad Fadzil M. II-694
Hasan, Mahmudul I-136
Hasan, Mohammad Khatim II-13, III-298
Hashim, Mazlan I-1054, III-410
Heng, Swee-Huay I-644
Hermann, Gabor III-872
Hernández, Emilio I-360
Hernández-Serrano, Juan II-526
Ho, Ying-Chin III-625
Hong, Youn-Sik II-778
Hsu, Ping-Hui I-871
Hu, Po II-514
Hu, Yincui III-1106
Huang, C.Y. I-767
Huang, Yu-Ying I-550
Huang, Yueh-Min I-1119
Huh, Eui-Nam II-383, II-455, II-1028
Huh, MoonHaeng I-1024
Hwang, Bo-Sung I-900
Hwang, EenJun III-968
Hwang, Hyun-Suk II-584
Hwang, Min-Shiang III-273
Hwang, Miyoung II-751
Hyun, Seunghwan I-252

Ibrahim, Hamidah III-115, I-748, III-763
Ida, Tetsuo II-653
Iglesias, Andrés II-680
Im, Eul Gyu II-75
Inceoglu, Mustafa Murat I-1133
Işlier, A. Attila I-886
Izquierdo, Antonio II-540

Jafari, Fahimeh III-398
Jameel, Hassan II-1028
Jang, Dae-Sik I-1074
Jang, Dong-Sik I-372
Jang, HaeSuk II-751
Jang, Ki-young II-275
Jang, Seokho III-486
Jang, Seong-Whan III-1076
Janin, Lilian III-154
Jeng, Yu-Lin I-1119
Jeon, Byung-gil I-238
Jeon, Jin-Oh I-634
Jeon, Sung-Hoon I-689
Jeon, Tae Gun II-584
Jeong, An Ryeol II-406
Jeong, Tae-Gun II-873, II-895
Jeong, Taikyeong I-1096
Jia, Xiaohua II-247
Jiawan, Zhang III-1056
Jie, Min Seok III-958
Jin, Seung Il III-844
Jizhou, Sun III-1056
John, Youngjun I-1096
Ju, Shiguang I-315
Jun, Eun A. II-113
Jun, SungIk I-655, I-912
Jung, Hoyoung I-224
Jung, Hyo Sang II-883
Jung, HyunRyong I-164

Jung, Jae-il II-122, II-131
Jung, JaeGyu I-164
Jung, Jaemin I-238
Jung, Keechul I-1063
Jung, Kwang-Mo II-593
Jung, Seok Won II-113
Jung, Seunghwan III-1141
Jung, Woo Jin II-216

Kacsuk, Peter III-872
Kagadis, George III-239, III-555
Kalatzis, Ioannis III-239, III-555, III-566
Kang, Euisun II-207, II-360
Kang, Euiyoung III-179
Kang, Hyunho III-1046
Kang, Jeong Seok I-346
Kang, Jeonil II-1085, III-886
Kang, Jinsuk I-1096
Kang, Mikyung III-169
Kang, Min-Sup I-634
Kang, MunSu II-397
Kang, OhHyung I-1024
Kang, Pilyong II-1018
Kang, S.H. I-562
Kang, Seon-Do I-372
Kang, Soojin II-951
Kang, Sooyong I-224
Karsak, E. Ertugrul I-779
Kasprzak, Andrzej III-343
Kendall, Graham III-611
Khader, Dalia III-1086
Khan, Adil Mehmood II-1028
Khan, Faraz Idris II-383, II-1028
Khattri, Sanjay Kumar I-525
Khonsari, Ahmad III-367, III-398
Ki, Hyung Joo II-216
Kim, Bo Hyun III-1
Kim, Byunggi II-751, II-764
Kim, C.W. II-961
Kim, Chang-Soo II-584
Kim, Chong-Min I-55, II-639
Kim, Daeyoung II-987
Kim, Deok-Soo I-55, II-639
Kim, Dojoong II-895
Kim, Dong-Hoi III-585
Kim, Donguk II-639
Kim, Doo-young II-122
Kim, Eun-ki I-238
Kim, Eunhoe I-325, II-839
Kim, Gui-Jung III-978, III-997
Kim, Gye-Young I-992, I-1003, I-1034, I-1045, I-1083
Kim, Gyoryeong I-1063
Kim, Hanil III-169
Kim, Hong-Yeon I-178
Kim, Hyojun I-164
Kim, Hyun-Ki III-1076
Kim, In Jung II-75
Kim, Jae-Yearn I-710
Kim, Jeom-goo II-113, I-140, II-148
Kim, Jeong Geun III-858, III-1017
Kim, Ji-Hong II-778
Kim, Jin III-585
Kim, Jin-Hyuk I-164
Kim, Jin Myoung III-637
Kim, Jin Ok III-790
Kim, Jong-Ki III-834
Kim, Jong-Myoung II-1107
Kim, Jongpil I-655
Kim, June I-178
Kim, Jungrae II-275
Kim, Ki-Sang I-1074
Kim, KiJoo II-397
Kim, KoonSoon III-886
Kim, Kuinam J. II-166, II-177, II-185
Kim, Kwang-Hoon III-900, III-910, III-920
Kim, Kyungjun III-201
Kim, Kyungwha I-189
Kim, Misun II-334
Kim, Miyoung II-479
Kim, Moon Jeong III-1187
Kim, Moonseong II-469, III-834
Kim, Myoung-Joon I-178
Kim, Sang-Tea III-950
Kim, Sang-Wook III-169
Kim, Sangjin II-987
Kim, Se-Hoon I-1014
Kim, Seok Tae II-157
Kim, Seong-Dong II-593
Kim, Seong Hoon I-346
Kim, Sun-Youb II-104
Kim, Sung-Min I-689
Kim, Tae Eun III-790
Kim, Won II-275, II-1072
Kim, Yong-Ho II-177
Kim, Yong-Hyun II-778
Kim, Yong-Ki III-1165
Kim, Young-Chang III-1165

Kim, Young-Kyun I-178
Kim, Young Yong I-622
Knizhnik, A. III-213
Ko, Il-Ju I-961
Ko, Kyong-Cheol I-1003
Ko, Kyounghee II-1018
Ko, Sung-Seok I-758
Ko, Sung Lim II-931
Ko, Young-Woong III-585
Ko, YunJung I-701
Kobayashi, Kingo III-1046
Koh, Kern I-213, I-252
Kondratenko, Yuriy I-305
Kong, Chunum II-1072
Koo, Gyodu II-415
Kook, Joong-Jin II-593
Kosloff, Todd J. III-1124
Kostopoulos, Spiros III-555
Kurkoski, Brian III-1046
Kwak, Donggyu II-818
Kwon, Oh-Cheon III-968
Kwon, Ohhoon I-213

Laganà, Antonio I-295
Larson, J. Walter III-55
Latip, Rohaya III-763
Lee, Amy H.I. I-767, I-860
Lee, Bohyung II-639
Lee, Bong-Hwan II-1097
Lee, Chang Ho III-33
Lee, Changwoo II-919
Lee, Chang Yong II-157
Lee, Cheol-Won II-75
Lee, Do-hyeon II-122, II-131, II-140, II-148
Lee, Dong Hoon III-1187
Lee, Dong Hwi II-166, II-177, II-185
Lee, Dongseob III-201
Lee, Gang Taek II-166
Lee, Geuk II-157
Lee, Gyu Bong III-1
Lee, Hae Young II-573
Lee, Hoonjae II-503, II-1008
Lee, Hun Soon III-844
Lee, Hyejeong I-201
Lee, Hyun Chan I-55
Lee, Ig-hoon I-189
Lee, Im-Yeong I-666
Lee, Jaeho II-706
Lee, Jangwon II-919
Lee, Jeog-bae II-140
Lee, Jongchan II-751, II-764, II-818
Lee, Joong jae I-992
Lee, Junghoon III-169, III-179
Lee, Kang Woong III-958
Lee, KeunSoo I-1034
Lee, Ki-Young II-778
Lee, Kwan-Soo II-951
Lee, Kwang Kook I-346
Lee, Kyu-Won II-1097
Lee, Kyunghye II-237, II-424, II-1052
Lee, Pill-Woo II-455
Lee, S.S. II-961
Lee, Sanggon II-503, II-1008
Lee, Sangho II-286
Lee, Sangmin I-178
Lee, Sehwan I-252
Lee, Seok-Lae I-900
Lee, Seok Cheol II-584
Lee, Seok Woo III-11
Lee, Seoung Soo II-883
Lee, SeungGwan III-140
Lee, SooCheol III-968
Lee, Sung-Hoon II-639
Lee, Sungchang II-469
Lee, Sung Hee III-20
Lee, Sungju III-1141
Lee, Tae-Jin II-216
Lee, Wan-Soo I-926
Lee, Wookey III-1007
Lee, Youho II-286
Lee, Yunli I-1063
León, Carlos III-811
Li, Chun-Ta III-273
Li, Gen III-353
Li, HaiMing II-434
Li, Kai II-434
Li, Sikun III-649
Li, Tiancheng III-1037
Li, Yaohang III-507
Liao, Gwo-Liang III-520
Lide, Fang I-462
Lim, Chungyu I-1063
Lim, Jong In II-113
Lim, Meng-Hui II-503, II-1008
Lim, Seung-Kil III-1007
Lim, Taesoo III-1007
Lim, YoungHwan II-207, II-360
Lin, Shih-Lin III-431
Lischka, Hans I-281

Liu, GuangLi III-1106
Liu, Hai II-247
Liu, YunLing III-1106
Liu, Zhihai I-678
Lopez, Javier II-549
López, Victoria López II-489
Luca, Adrian III-460
Luo, Dongwei II-559
Luo, Yanmin III-821
Luo, Ying III-1106

Ma, Yuanchen II-514
Maarof, Mohd Aizaini I-512
Maheshwari, Anil I-82
Mahmud, Ahmad Rodzi III-128
Malamas, Menelaos III-239
Malik, Ariff Md Ab III-611
Mamat, Ali III-115
Mansor, Shattri III-128
Manulis, Mark I-603
Mao, Chengying III-92
Mao, Jane I-589
Marghany, Maged I-1054, III-410
Marin, Mircea II-653
Marin, Rafa II-1038
Martono, Wahyudi II-85
Mascagni, Michael III-507
Matos, Inês I-1
Meng, Xiangxu I-110
Mesgari, Saadi II-308
Michaelis, Bernd I-397
Miksch, Silvia III-660
Millan, Marta II-370
Min, SangWon II-593
Min, Seungki I-1063
Mitrea, Adrian I-409
Mitrea, Mihai I-409, III-445
Mo, Eun Jong III-958
Mohaisen, Abedelaziz III-886
Mohamed, Azlinah I-576
Moon, Dae-sung III-1141, I-1146
Moon, Hyun-Joo II-764, II-849
Moon, Jong-Sik I-666
Moon, Jongbae II-818
Morgado, José II-666
Mu, Yi III-1096
Muñoz, Andrés III-710
Muda, Azah Kamilah I-385
Mukhopadhyay, Asish I-41
Mun, Youngsong II-226, II-237, II-334, II-415, II-424, II-479, II-1052, II-1062
Mutalib, Sofianita I-576

Na, Yang II-907
Nazr-e-Batool II-694
Niese, Robert I-397
Nikiforidis, George III-239, III-555, III-566
Noh, Minki II-415
Norris, Boyana III-55
Nouri, Mostafa I-68
Nourollah, Ali I-15
Nussbaum, Doron I-82
Nyang, DaeHun II-1085, III-886
Nyman, Gunnar III-1114

Ogryczak, Włodzimierz I-804
Oh, Heekuck II-987
Oh, Jung Min II-157
Oh, Kie-Sung I-1014
Oh, KyoungSu I-972
Oh, Sanghyeok I-1063
Oh, Sung-Kwun III-1076
Ohbo, Nobuo III-821
Onieva, Jose A. II-549
Ortiz, Edward II-370
Othman, Mohamed II-13, II-261, III-298, I-748, III-763
Othman, Zulaiha Ali III-248

Pacheco, Vinícius II-790
Pai, Ping-Feng I-550
Pan, Alberto II-322
Pan, Jingui I-315
Panning, Axel I-397
Panshenskov, Mikhail II-38
Pardede, Eric III-749
Park, C.H. II-961
Park, DaeHyuck II-207, II-360
Park, Gyung-Leen III-169, III-179, I-1096
Park, Hong Seong I-346
Park, Hyoung-Keun II-104
Park, Jea Han I-346
Park, Jeonghoon II-286
Park, Jin-Young I-1045
Park, Joon Young II-706
Park, Keon-Jun III-1076
Park, Keun Sang II-883

Park, KiHong I-1024
Park, Kisoeb III-834
Park, Miae III-1141
Park, Min-Jae III-900
Park, Rhohun I-55
Park, Sang-Jo I-634
Park, Sangho III-486
Park, Sangjoon II-751, II-764, II-818
Park, Seon-Ho II-157, II-195, II-1107
Park, Sung Bum III-1
Park, Sungmin I-224
Park, SunYong I-972
Park, Youngho II-503, II-1008
Pastor, Rafael I-791
Pazos, R. Rodolfo A. I-424, III-674
Pedraza, S. II-727
Pedrycz, Witold III-1076
Pegueroles, Josep II-526
Pei, Zheng II-434
Pérez O., Joaquín I-424, III-674
Petrescu, Andrei D. I-450
Petrović, Slobodan III-597
Phan, Ca Van III-858
Phan, Raphael C.-W. I-951
Piattini, Mario III-262
Poch, Jordi II-861
Pooyandeh, Majeed II-308
Porrini, Massimiliano I-295
Porschen, Stefan I-96
Potapkin, B. III-213
Potdar, Vidyasagar III-227, III-724
Prados, Ferran II-727, II-861
Prats, A. II-727
Prêteux, F. III-445
Przewoźniczek, Michał III-330
Puig, J. II-727
Puig-Pey, Jaime II-680
Puttini, Ricardo II-790

Qi, Meng I-110
qiang, Chen I-469
Qin, Zheng III-1066
Qu, Rong III-611

Ra, Yu-Chan II-104
Raazi, S.M.K. II-1028
Radi, Mohammed III-115
Rahayu, J. Wenny III-749
Rahim, Mohd Shafry Mohd III-128
Rahman, Md Mizanur II-25
Rahman, Shuzlina Abdul I-576
Ramadan, Omar III-388, III-421
Ramos, J.I. III-941
Rao, S.V. I-41
Raposo, Juan II-322
Razzazi, Mohammad Reza I-15
Rendell, Alistair P. I-265
Reyes S., Gerardo III-674
Rhee, YangWon I-1024
Rivera-López, Rafael III-697
Robinson, Ian III-1037
Rodionov, Alexey S. III-534
Rodionova, Olga K. III-534
Rodrigues, Rui II-666
Rokne, Jon G. I-136
Romero, L.F. III-941
Romoozi, Morteza II-804
Ropero, Jorge III-811
Rosado, David G. III-262
Ruckenbauer, Matthias I-281
Rughooputh, S.D.D.V. III-774
Ryba, Przemyslaw III-343
Ryu, Ho Yeon II-883
Ryu, Joonghyun I-55, II-639
Ryu, Kwang Yeol III-11
Ryu, Seonggeun II-1052, II-1062
Ryu, Su-Bong I-634
Ryu, Yeonseung I-201, I-213

Sack, Jörg-Rüdiger I-82
Sadoghi, H. III-495
Safaei, F. III-367
Salama, Rafik A. I-536
Salami, Momoh Jimoh E. II-85
Salavert, Isidro Ramos III-104
Saleh, Moutaz III-248
Sameh, Ahmed I-536
Sanabani, M. II-261
Sanjay, Kumar Khattri I-305
Santa, José III-710
Sbert, Mateu II-602, II-741
Schwandt, Hartmut III-285
Schwenk, Jörg I-603
Sellarès, J. Antoni I-26
Semé, David III-379
Seo, Jeongyeon I-55
Seo, Jungtaek II-94
Seok, Bo-Hyun II-455
Shad, Rouzbeh II-308
Shamala, Subramaniam III-115, II-261

Shamsuddin, Siti Mariyam I-385, I-512
Shariff, Abdul Rashid Mohamed III-128
Shen, Shengyu III-649
Shiau, J.Y. I-721
Shim, Byoung-Sup II-104
Shim, Hyoki I-224
Shim, Junho I-189
Shin, Dong Keun II-931
Shin, Hyungjong I-238
Shin, In-Hye III-169
Shin, Keehyun II-919
Shin, Kyoungho I-335, II-829
Shin, SeongYoon I-1024
Shin, Teail II-1052
Shin, Woo Jin III-1017
Sierra, José María II-540, II-549
Sifaki, Koralia III-239
Silva, Frutuoso II-666
Sim, Wontae II-1018
Skarmeta, Antonio F.G. III-710
Śliwiński, Tomasz I-804
Soler, Josep II-861
Solomou, Ekaterini III-239
Song, Hyoung-Kyu III-950
Song, Jiyoung II-764
Song, Joo-Seok I-900
Song, Joon-Yub III-486
Song, Kiwon II-996
song, Wanqing I-469
Soriano, Miguel II-526
Sterian, Andreea Rodica I-436, I-450
Sterian, Paul E. I-450
Subramaniam, S. I-748
Subramaniyam, Murali III-486
Sulaiman, Jumat II-13, III-298
Sulaiman, Md Nasir III-763
Sun, RuiZhi III-1106
Sun, Xingming III-1066
Susilo, Willy III-1096

Tabik, S. III-941
Tae, Kang-Soo II-406, II-455
Takahashi, Hidekazu II-653
Talebanfard, N. III-367
Talebi, Mohammad S. III-398
Tang, Chuan Yi III-520
Taniar, David III-749
Teng, Hui-Ming I-871
Tet-Khuan, Chen I-151
Theera-Umpon, Nipon III-190
Ting, Grace C.-W. I-644
To, Hoang Minh II-931
Tolga, Ethem I-832
Toropov, Andrey II-941
Torres, Joaquin II-540
Tran, Minh II-627
Trofimov, Vadim I-122
Trujillo, Maria II-370
Tsai, Chieh-Yuan I-1107
Tseng, Y.F. I-562, I-721
Tumin, Sharil I-305
Tung, Pi-Cheng III-431
Tzeng, Shiang-Feng III-273

Uhmn, Saangyong III-585
Ulutaş, Berna Haktanirlar I-886
Um, Sukkee II-951

Vakhitov, Alexander II-38
Valuev, I. III-213
Villalba, L. Javier García II-489
Vlad, Adriana I-409, III-445, III-460
Vyatkina, Kira I-122

Walkowiak, Krzysztof III-319, III-330
Wang, Bin III-1066
Wang, Huanzhao II-559
Wang, Jianqin III-1106
Wang, Jiaye I-110
Wang, Kun-Te I-1119
Wang, Tzone-I I-1119
Wang, Xiaoting I-110
Wee, Hui-Ming I-562, III-625, I-721, I-734, I-871
Weon, Sunhee I-1034
Won, Chung-In II-639
Won, Dongho II-996
Won, Jae-Kang III-910
Won, Youjip I-238
Wongthongtham, Pornpit II-346
Wu, Chaolin III-1106
wu, Chenchen III-1152
Wu, Qianhong III-1096

Xiang, Dan II-434
Xiwen, Li I-462
Xu, Dan I-315
Xu, Lu III-69
Xu, Qing II-602, II-741

Xue, Lingzhou I-678
Xue, Yong III-1106

Yaghmae, Mohammad H. III-398
Yamaguchi, Kazuhiko III-1046
Yang, Chenglei I-110
Yang, Gao III-1056
Yang, Hengfu III-1066
Yang, Hyosik I-1096
yang, Jianguo I-469
Yang, P.C. I-562, I-721, I-734
Yang, Rui I-265
Yang, Xuejun III-353
Yanrong, Pang I-462
Yap, Vooi Voon II-694
Yau, Wei Chuen II-975
Ye, Letian I-678
Yeh, C.-T. I-937
Yi, Do Won II-157
Yi, Okyeon III-1141
Yildiz, Burcu III-660
Yim, Changhoon I-622
Yoo, Hun-Woo I-372
Yoo, Jae-Woo II-849
Yoo, Ji Ho III-1017
Yoo, Kee-Young I-926
Yoo, Sehwan I-1096
Yoo, Yun-Seok I-201
Yoon, DaeSub III-968
Yoon, Eun-Jun I-926
Yoon, Kyunghoon I-224
Yoon, Taehyun I-1063
You, Kang Soo II-406
You, Young-Hwan III-950
Youlou, Sidney III-379
Yu, Jonas C.P. I-734, I-818
Yu, KiSung II-415
Yu, Seung-Hwan II-951
Yuan-jun, He II-716
Yusoff, Marina I-576

Zahara, E. I-562
Zainal, Anazida I-512
Zapata, Santiago II-1038
Zarate, M. Jose A. I-424
Zarei, Alireza I-68
Zhang, Fanguo III-1096
Zhang, Jianfeng II-602
Zhang, Jianhong I-589, II-63
Zhang, Jianmin III-649
Zhang, Liguo I-480, III-69
Zhang, Ying III-353
Zhang, Yu II-612
Zhao, Qingping II-559
Zheng, Lei III-1106
Zhi-ping, Hu II-716
Zhou, Jianying II-549
Zhou, Jin III-1056
Zlatanova, Sisi I-151
Zong-ying, Ou II-716
Zukarnain, Z. II-261