Lecture Notes in Computer Science 2864
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Anind K. Dey   Albrecht Schmidt
Joseph F. McCarthy (Eds.)

UbiComp 2003: Ubiquitous Computing

5th International Conference
Seattle, WA, USA, October 12-15, 2003
Proceedings

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Anind K. Dey
Intel Research Berkeley
2150 Shattuck Ave, Suite 1300, Berkeley, CA 94704, USA
E-mail: anind@intel-research.net

Albrecht Schmidt
Ludwig-Maximilians-Universität München, Institut für Informatik
LFE Medieninformatik, Arcisstr. 21, 90290 München, Germany
E-mail: albrecht.schmidt@informatik.uni-muenchen.de

Joseph F. McCarthy
Intel Research Seattle
1100 NE 45th St., 6th Floor, Seattle, WA 98105, USA
E-mail: mccarthy@intel-research.net

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

CR Subject Classification (1998): C.2, C.3, D.2, D.4, H.4, H.5, K.4

ISSN 0302-9743
ISBN 3-540-20301-X Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 10966358 06/3142 5 4 3 2 1 0

Preface

UbiComp 2003, the 5th Annual Conference on Ubiquitous Computing, is the premier forum for presentation of research results in all areas relating to the design, implementation, deployment and evaluation of ubiquitous computing technologies. The conference brings together leading researchers, from a variety of disciplines, perspectives and geographical areas, who are exploring the implications of computing as it moves beyond the desktop and becomes increasingly interwoven into the fabrics of our lives.

This volume, the conference proceedings, contains the entire collection of high-quality full papers and technical notes from UbiComp 2003. There were 16 full papers in this year's conference, selected by our international program committee from among 117 submissions. There were also 11 technical notes in the program: 8 taken from the 36 technical note submissions, and 3 that were adapted from full paper submissions. We are very grateful to Tim Kindberg and Bernt Schiele, our Technical Notes Co-chairs, and to all the authors and reviewers of both papers and technical notes who contributed to maintaining the high standards of quality for the conference.

In addition to the full papers and technical notes, UbiComp 2003 also provided a number of other participation categories, including workshops, demonstrations, interactive posters, a panel, a doctoral colloquium and a video program. While accepted submissions in these categories were also of high quality, timing and size constraints prevented us from including them in this volume. They were instead available in a printed conference supplement distributed to conference attendees and at the conference Web site, www.ubicomp.org. We also want to express our thanks to all the chairpersons, authors and reviewers in these categories for their contributions to making the conference such an outstanding success.

Several organizations provided financial and logistical assistance for the conference, and we are grateful for their support. The donations by our corporate benefactor, Intel, and by our corporate sponsors, Fuji Xerox Palo Alto Laboratory, Hewlett-Packard Laboratories, Microsoft Research, Nokia Research, and Smart Technologies, help us provide a world-class conference experience for all attendees. We are also grateful to Intel Research Berkeley and Intel Research Cambridge for providing the spaces and other logistical support for our dual-site transcontinental program committee meeting in June, and to Microsoft Research for the use of their Conference Management Toolkit.

Finally, we wish to thank all the people attending the conference, as it is the opportunities to meet and interact with all of you interesting people that makes the planning of such a momentous event a worthwhile endeavor for all involved!

August 2003

Joe McCarthy
Anind K. Dey
Albrecht Schmidt

Conference Organization

Organizers

Conference Chair	
Joe McCarthy	Intel Research Seattle (USA)
Program Chairs	
Anind K. Dey	Intel Research Berkeley (USA)
Albrecht Schmidt	University of Munich (Germany)
Technical Notes Chairs	
Tim Kindberg	Hewlett-Packard Labs (USA)
Bernt Schiele	ETH Zurich (Switzerland)
Demonstrations Chairs	
Eric Paulos	Intel Research Berkeley (USA)
Allison Woodruff	Palo Alto Research Center (USA)
Interactive Posters Chairs	
Marc Langheinrich	ETH Zurich (Switzerland)
Yasuto Nakanishi	University of Electro-Communications (Japan)
Videos Chairs	
Peter Ljundstrand	PLAY, Interactive Institute (Sweden)
Jason Brotherton	Ball State University (USA)
Doctoral Colloquium Chair	
Tom Rodden	Nottingham University (UK)
Workshops Chair	
Michael Beigl	TecO, University of Karlsruhe (Germany)
Panels Chair	
Gerd Kortuem	Lancaster University (UK)
Student Volunteers Chair	
Stephen Voida	Georgia Institute of Technology (USA)
A/V & Computing Chair	
James Gurganus	Intel Research (USA)
Treasurer	
David McDonald	University of Washington (USA)
Publications Chair	
James Scott	Intel Research Cambridge (UK)
Publicity Chair	
Mike Hazas	Lancaster University (UK)
Webmaster	
Khai Truong	Georgia Institute of Technology (USA)
Local Arrangements	
Ellen Do	University of Washington (USA)
Conference Manager	
Debra Bryant	University of Washington (USA)

Sponsors

Corporate Benefactor	Intel
Corporate Sponsors	Fuji Xerox Palo Alto Laboratory Hewlett-Packard Laboratories Microsoft Research Nokia Research Smart Technologies

Supporting Societies

UbiComp 2003 enjoys in-cooperation status with the following special interest groups of the Association for Computing Machinery (ACM):

SIGCHI (Computer-Human Interaction)
SIGSOFT (Software Engineering)
SIGMOBILE (Mobile Computing) - pending

Program Committee

Gregory Abowd	Georgia Institute of Technology (USA)
John Barton	Hewlett-Packard Labs (USA)
Barry Brumitt	Microsoft (USA)
George Coulouris	Cambridge University (UK)
Joelle Coutaz	CLIPS-IMAG (France)
James L. Crowley	I.N.P. Grenoble, INRIA Rhône-Alpes (France)
Nigel Davies	Lancaster University (UK), University of Arizona (USA)
Anind K. Dey	Intel Research Berkeley (USA)
Paul Dourish	University of California at Irvine (USA)
Keith Edwards	Palo Alto Research Center (USA)
Alois Ferscha	Universität Wien (Austria)
Armando Fox	Stanford University (USA)
Hans-Werner Gellersen	Lancaster University (UK)
Beverly Harrison	IBM Research (USA)
Lars Erik Holmquist	Viktoria Institute (Sweden)
Eric Horvitz	Microsoft Research (USA)
Antonio Krüger	Saarland University (Germany)
Spyros Lalis	ICS-FORTH (Greece)
Kenji Mase	Nagoya University, ATR (Japan)
Joe McCarthy	Intel Research Seattle (USA)
Chandra Narayanaswami	IBM Research (USA)
Tom Rodden	Nottingham University (UK)
Yvonne Rogers	Sussex University (UK)
Chris Schmandt	MIT Media Lab (USA)

Program Committee (contd.)

Albrecht Schmidt	University of Munich (Germany)
Howard Shrobe	MIT AI Lab (USA)
Itiro Siio	Tamagawa University (Japan)
Mark Smith	Hewlett-Packard Labs (USA)
Yoichi Takebayashi	Shizuoka University (Japan)
Roy Want	Intel Research (USA)

Reviewers

Mark Ackerman	University of Michigan (USA)
John Ankcorn	HP Labs (USA)
Daniel Ashbrook	Georgia Institute of Technology (USA)
Chris Atkeson	Carnegie Mellon University (USA)
Yuji Ayatsuka	Sony CSL (Japan)
Rajesh Balan	Carnegie Mellon University (USA)
Rafael Ballagas	Stanford University (USA)
Thomas Barkowsky	University of Bremen (Germany)
Dominique Barthel	France Telecom R&D (France)
Len Bass	Carnegie Mellon University (USA)
Patrick Baudisch	Microsoft Research (USA)
Martin Bauer	University of Stuttgart (Germany)
Jörg Baus	Saarland University (Germany)
Michel Beaudouin-Lafon	Université Paris-Sud (France)
Wolfgang Beer	University of Linz (Austria)
Bo Begole	Sun Microsystems (USA)
Michael Beigl	University of Karlsruhe (Germany)
Genevieve Bell	Intel Research (USA)
Steve Benford	University of Nottingham (UK)
Francois Berard	Grenoble Technological University (France)
Alastair Beresford	Cambridge University (UK)
Stefan Berger	IBM Research (USA)
Jan Beutel	ETH Zurich (Switzerland)
Mark Billinghurst	University of Canterbury, HIT Lab NZ (New Zealand)
Thomas Binder	Interactive Institute (Sweden)
Günther Blaschek	University of Linz (Austria)
Jeanette Blomberg	IBM Research (USA)
Prasad Boddupalli	University of Arizona (USA)
Philippe Bonnet	University of Copenhagen (Denmark)
Jan Borchers	ETH Zurich (Switzerland)
Paolo Busetta	ITC-IRST (Italy)
Andreas Butz	Saarland University (Germany)
Ramon Caceres	IBM Research (USA)

Reviewers (contd.)

Vinny Cahill	Trinity College Dublin (Ireland)
Ozan Cakmakci	Center for Research in Optics and Lasers, Orlando (USA)
Chris Campbell	IBM Research (USA)
Roy Campbell	UIUC (USA)
John Canny	UC Berkeley (USA)
Matthew Chalmers	University of Glasgow (UK)
Guanling Chen	Dartmouth College (USA)
Keith Cheverst	Lancaster University (UK)
Volker Christian	University of Linz (Austria)
Brent Chun	Intel Research (USA)
Sunny Consolvo	Intel Research (USA)
Andy Crabtree	University of Nottingham (UK)
Edward Cutrell	Microsoft Research (USA)
Mary Czerwinski	Microsoft Research (USA)
Trevor Darrell	MIT (USA)
Randall Davis	MIT AI Lab (USA)
Fiorella de Rosis	University of Bari (Italy)
Prasun Dewan	North Carolina State University (USA)
Alan Dix	Lancaster University (UK)
Lance Doherty	University of California (USA)
Mark Dunlop	University of Strathclyde (UK)
Maria Ebling	IBM Research (USA)
Markus Eisenhauer	Fraunhofer Gesellschaft (Germany)
Jennica Falk	Viktoria Institute (Sweden)
Shelly Farnham	Microsoft Research (USA)
John Fawcett	Cambridge University (UK)
Christer Fernstrom	Xerox Research Centre Europe (France)
Sebastian Fischmeister	University of Salzburg (Austria)
Ken Fishkin	Intel Research (USA)
George Fitzmaurice	Alias (SGI) (Canada)
Geraldine Fitzpatrick	University of Sussex (UK)
Kris Fleming	Intel Research (USA)
Harold Fox	MIT AI Lab (USA)
Steve Freeman	Independent Consultant (UK)
Adrian Friday	Lancaster University (UK)
Krzysztof Gajos	University of Washington (USA)
David Garlan	Carnegie Mellon University (USA)
Lalya Gaye	Viktoria Institute (Sweden)
Vadim Gerasimov	MIT Media Lab (USA)
Stuart Golden	Intel Research (USA)
Rich Gossweiler	HP Labs (USA)
Steven Gribble	University of Washington (USA)
Robert Grimm	New York University (USA)

Reviewers (contd.)

Rebecca Grinter	Palo Alto Research Center (USA)
Jonathan Grudin	Microsoft Research (USA)
Indranil Gupta	Cornell University (USA)
Maria Håkansson	Viktoria Institute (Sweden)
Rob Harle	Cambridge University (UK)
Koiti Hasida	National Institute of Advanced Industrial Science and Technology (AIST) (Japan)
Paul Havinga	University of Twente (Netherlands)
Gillian Hayes	Georgia Institute of Technology (USA)
Mike Hazas	Lancaster University (UK)
Jeff Hightower	University of Washington (USA)
Ken Hinckley	Microsoft Research (USA)
Jim Hollan	UC San Diego (USA)
Simon Holland	Open University (UK)
Jason Hong	UC Berkeley (USA)
Andy Hopper	Cambridge University (UK)
Marc Huber	Intelligent Reasoning Systems (USA)
Scott Hudson	Carnegie Mellon University (USA)
Richard Hull	HP Labs (UK)
Jan Humble	SICS (Sweden)
Stephen Intille	MIT (USA)
Arna Ionescu	Stanford University (USA)
Shahram Izadi	University of Nottingham (UK)
Robert Jacob	Tufts University (USA)
Brad Johanson	Stanford University (USA)
Matt Jones	University of Waikato (New Zealand)
Anupam Joshi	University of Maryland (USA)
Oskar Juhlin	Interactive Institute (Sweden)
Hiroshi Kanazawa	Toshiba (Japan)
Ernst Kashofer	University of Linz (Austria)
Masaji Katagiri	DoCoMo Labs (USA)
Kazuhiko Kato	Tsukuba University (Japan)
Tetsuya Kawahara	Kyoto University (Japan)
Rick Kazman	Carnegie Mellon University (USA)
Pedram Keyani	Stanford University (USA)
Graham Kirby	St Andrews University (UK)
Scott Klemmer	UC Berkeley (USA)
Tetsunori Kobayashi	Waseda University (Japan)
Kimberle Koile	MIT AI Lab (USA)
Boriana Koleva	University of Nottingham (UK)
Fabio Kon	University of São Paulo (Brazil)
Shin'ichi Konomi	University of Colorado at Boulder (USA)
Gerd Kortuem	Lancaster University (UK)
Gabriele Kotsis	University of Linz (Austria)

Reviewers (contd.)

David Kotz	Dartmouth College (USA)
Sacha Krakowiak	Université Joseph Fourier, INRIA Rhône-Alpes (France)
Christian Kray	Lancaster University (UK)
Arvind Krishnamurthy	Yale (USA)
Peter Kropf	University of Montreal (Canada)
John Krumm	Microsoft Research (USA)
Uwe Kubach	SAP Corporate Research (Germany)
Anthony LaMarca	Intel Research (USA)
Mark Langheinrich	ETH Zurich (Switzerland)
Seon-Woo Lee	Hallym University (Korea)
Scott Lee-Tiernan	Microsoft Research (USA)
Toby Lehman	IBM Research (USA)
Hui Lei	IBM Research (USA)
John Light	Intel Research (USA)
Ben Ling	Stanford University (USA)
Jie Liu	Palo Alto Research Center (USA)
Sara Ljungblad	Viktoria Institute (Sweden)
Gary Look	MIT AI Lab (USA)
Diego López de Ipiña	3G Labs (UK)
Rose Luckin	University of Sussex (UK)
Paul Luff	King's College (UK)
Lu Luo	Carnegie Mellon University (USA)
Kent Lyons	Georgia Institute of Technology (USA)
Alan Maclean	Image Semantics (UK)
Sam Madden	UC Berkeley (USA)
Ken-ichi Maeda	Toshiba (Japan)
Jennifer Mankoff	UC Berkeley (USA)
Gurmeet Manku	Stanford University (USA)
Natalia Marmasse	MIT Media Lab (USA)
Stefan Marti	MIT Media Lab (USA)
Friedemann Mattern	ETH Zurich (Switzerland)
Rene Mayerhofer	University of Linz (Austria)
David McGee	Pacific Northwest Laboratories (USA)
Malena Mesarina	HP Labs (USA)
Brian Meyers	Microsoft Research (USA)
Michihiko Minoh	Kyoto University (Japan)
Scott Mitchell	Apama Ltd. (UK)
David Morse	Open University (UK)
Max Mühlhäuser	University of Darmstadt (Germany)
Henk Muller	Bristol University (UK)
Elizabeth Mynatt	Georgia Institute of Technology (USA)
Tatsuo Nakajima	Waseda University (Japan)
Yasuto Nakanishi	University of Electro-Communications (Japan)

Reviewers (contd.)

Jin Nakazawa	Keio University (Japan)
Wolfgang Narzt	University of Linz (Austria)
William Newman	Independent Consultant (UK)
David Nguyen	Intel Research (USA)
Daniela Nicklas	University of Stuttgart (Germany)
Paddy Nixon	University of Strathclyde (UK)
Brian Noble	University of Michigan (USA)
Hisao Nojima	NTT (Japan)
Moira Norrie	ETH Zurich (Switzerland)
Kazushige Ouchi	Toshiba (Japan)
Leysia Palen	University of Colorado (USA)
Trevor Pering	Intel Research (USA)
Stephen Peters	MIT AI Lab (USA)
Ted Phelps	University of Sussex (UK)
Jeff Pierce	Georgia Institute of Technology (USA)
Gustav Pomberger	University of Linz (Austria)
Shankar Ponnekanti	Stanford University (USA)
Joachim Posegga	SAP Corporate Research (Germany)
Gilles Privat	France Telecom R&D (France)
Kevin Quigley	MIT AI Lab (USA)
MT Raghunath	IBM Research (USA)
Rakhi Rajani	HP Labs (USA)
Cliff Randell	University of Bristol (UK)
Anand Ranganathan	UIUC (USA)
Kumar Ranganathan	Intel CTG (USA)
Martin Raubal	University of Münster (Germany)
Kasim Rehman	Cambridge University (UK)
Patrick Reignier	Université Joseph Fourier (France)
Jun Rekimoto	Sony CSL (Japan)
Jochen Renz	TU Vienna (Austria)
Jon Rimmer	University of Sussex (UK)
Manuel Roman	DoCoMo Labs (USA)
Mark Rouncefield	Lancaster University (UK)
Larry Rudolph	MIT LCS (USA)
Umar Saif	MIT LCS (USA)
Koichi Sakurai	Kyusyu University (Japan)
Daniel Salber	IBM Research (USA)
Kam Sanmugalingam	Cambridge University (UK)
Johan Sanneblad	Viktoria Institute (Sweden)
Ichiro Satoh	National Institute of Informatics (Japan)
Bernt Schiele	ETH Zurich (Switzerland)
Bill Schilit	Intel Research (USA)
Jochen Schiller	Freie Universität Berlin (Germany)
Christoph Schlieder	University of Bamberg (Germany)

Reviewers (contd.)

Dieter Schmalstieg	TU Vienna (Austria)
James Scott	Intel Research (UK)
Jochen Seitz	Technical University Illmenau (Germany)
Abigail Sellen	HP Labs (UK)
Steven Shafer	Microsoft Research (USA)
Gurdip Singh	Kansas State University (USA)
Tobias Skog	Viktoria Institute (Sweden)
Diana Smetters	Xerox PARC (USA)
Marc Smith	Microsoft Research (USA)
Dave Snowdon	Xerox Research Centre Europe (France)
Rip Sohan	University of Cambridge (UK)
Mirjana Spasojevic	HP Labs (USA)
Peter Steenkiste	Carnegie Mellon University (USA)
Paul Stewart	Palo Alto Research Center (USA)
Ivan Stojmenovic	University of Ottawa (Canada)
Thomas Stricker	ETH Zurich (Switzerland)
Mark Stringer	Cambridge University (UK)
Yasuyuki Sumi	Kyoto University/ATR (Japan)
Murali Sundar	Intel Research (USA)
Dan Swinehart	Palo Alto Research Center (USA)
Desney Tan	Carnegie Mellon University (USA)
Yoshito Tobe	Tokyo Denki University (Japan)
Peter Tolmie	Rank Xerox Research Center (France)
Kentaro Toyama	Microsoft Research (USA)
Jonathan Trevor	FX Palo Alto Laboratory (USA)
Joe Tullio	Georgia Institute of Technology (USA)
Brygg Ullmer	Konrad Zuse Center for Information Technology (Germany)
Sunil Vemuri	MIT Media Lab (USA)
Simon Vogl	University of Linz (Austria)
Stephan Voida	Georgia Institute of Technology (USA)
Andy Ward	Ubiquitous Systems Limited (UK)
Sanjiva Weerawarana	IBM Research (USA)
Pierre Wellner	IDIAP (Switzerland)
Alan Wexelblat	HOVIR (USA)
Andy Wilson	Microsoft Research (USA)
Allison Woodruff	Palo Alto Research Center (USA)
Michiaki Yasumura	Keio University (Japan)
Adel Youssef	Microsoft Research (USA)
Bin Zhang	ATR (Japan)
Ingrid Zukerman	Monash University (Australia)

Table of Contents

Location and Space

Modeling and Inference

Context Awareness

New Devices and Technologies

Domestic Environments and Healthcare

Social Aspect and Privacy

New Interfaces

Building World Models by Ray-Tracing within Ceiling-Mounted Positioning Systems

Robert K. Harle and Andy Hopper

Laboratory for Communication Engineering
University of Cambridge
{rkh23,hopper}@eng.cam.ac.uk

Abstract. Context-aware computing in location-aware environments demands the combination of real world position with a computational world model to infer context. We present a novel approach to building world models using signals inherent in positioning systems, building on the work of the robotics research field.
We implement the approach using the Bat ultrasonic location system. We observe excellent results when trying to extract the height and shape of horizontal surfaces, and demonstrate how to image and characterise object volumes.
Results are collected using personnel Bats and by using an autonomous vehicle which moves randomly. In both cases, the results are accurate and reliable.

1 Introduction

The context-aware computing paradigm [14] has sparked an interest in the development of technologies that realise the applications it offers. Thus far, the major enabling technology has emerged as location (both through absolute positioning and spatial containment), with many research systems already demonstrated to locate personnel and objects [7, 17, 13, 15, 18, 4, 11, 10], and many more on the horizon.

Context-aware systems must integrate location information with a knowledge of the world in which they operate. This knowledge is contained within a world model, and may be as detailed or sparse as the applications which utilise it demand. As a minimum, the authors have found by experience that a useful system should be aware of room bounds, computer host positions, and the location of major office furniture. These facilitate the majority of applications such a hot-desking and "follow-me" applications.

The experiences of the authors with the SPIRIT context-aware system [1], have highlighted a series of practical problems that regularly manifest in a real world deployment. Objects modelled by SPIRIT, but not explicitly tracked by sensor systems, are continually observed to unexpectedly move, reconfigure, or be removed altogether. As an example, users commonly shift and re-orient their desktop display.

The resultant loss of synchronisation between the real world and the world model can be problematic for context-aware systems. If a monitor is moved, for example, the ability to hot-desk to it whilst in its vicinity is immediately lost. The system continues to make decisions consistent with its model, but now inconsistent with the real world. Users become confused, and start to distrust the system, reducing its value.

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 1–17, 2003.

To prevent this, context-aware systems must be able to respond to changes in the real world. Current implementations use a world model that is manually configured at initial switch on. Adaptation to any subsequent changes in the real world relies on human administrators observing those inconsistencies and taking the time to correct them. On the scale of a laboratory implementation, this is adequate. However, larger deployments clearly cannot depend on this approach.

This paper presents a method of using signals within positioning systems to infer the presence, position, and shape of objects in the environment with minimal human involvement.

2 Robotics and Positioning Signal Propagation

Unobtrusive positioning systems rely on the propagation of wave phenomena. The characteristics of the propagation (such as time-of-flight) or the wave itself (phase) provide information that can be combined with the information from other waves to compute a position. For example, the Bat system [17] uses the time-of-flight measurements of ultrasonic waves propagating between a transmitter and multiple, ceiling-mounted, receivers to perform a multilateration calculation and estimate the transmitter position.

Given the position of a transmitter and a receiver that detects an emitted signal, we establish a vector between these positions, known as a *ray*. Herein, it will be assumed that the positioning system uses a mobile transmitter and fixed receivers. All concepts are directly applicable to the inverse situation by symmetry arguments. As the transmitter is moved around, a series of new rays are established with each successful positioning calculation.

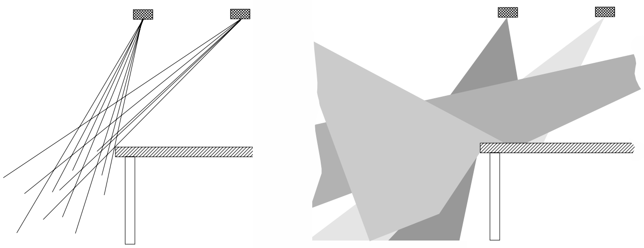


Fig. 1. The build up of rays around a table.

Over time, the rays penetrate into three-dimensional space. Where objects made of material that is opaque to the positioning medium exist, no rays are expected (illustrated in Fig. 1). The essential premise, then, is to facilitate the aggregation up of rays within an environment, and provide an analysis algorithm to extract information from the regions that no rays intercept.

In a real system, rays may build up very quickly. A three-dimensional position requires a minimum of three receiver sightings, and hence provides at least three rays. At a conservative update rate of 0.1Hz, we generate a minimum of 25,920 rays per person

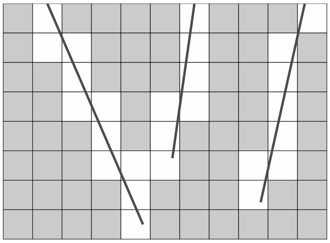


Fig. 2. A vertical slice through an occupancy grid. Grey cells are occupied. Rays are represented by thick dark lines.

per day. Robust systems are likely to vastly exceed this minimum. It is therefore not storage efficient to store the details of every ray within the system.

Instead, we can use a three-dimensional *occupancy grid*. Occupancy grids were first proposed by Moravec and Elfes in the field of robotics and autonomous navigation, where robots must use attached sensors to model their environment. There exists extensive literature in the field pertaining to mapping and exploration [9, 2, 8, 3, 16].

For the purposes of this paper, an occupancy grid is a three dimensional construction that segments the space of interest into regular cubes. Each cube is associated with a binary state from the set {occupied, unoccupied}.

A ray is quantised onto a grid by determining the cells it intercepts, and assigning them the unoccupied state (Fig. 2). Using a spatial grid reduces the storage requirements, and eases analysis. The accuracy of such a representation depends on the choice of cell size. A smaller cell size necessitates larger storage requirements, but potentially gives increased accuracy.

3 Real World Difficulties

Ideally a ray-tracing system would record the receivers that receive any positioning pulse and store the corresponding rays in some form. However, to do so would make the implicit assumption that the path traversed by the positioning signal was straight and direct between the transmitter and receiver. In reality, waves are subject to a series of physical phenomena which may cause deviations from this ideal. In particular, waves may be diffracted (Fig. 3(a)) or reflected (Fig. 3(b)).

The possibility of signals taking non-direct paths has traditionally been a serious problem for positioning systems. It introduces a source of significant error into the positioning algorithm. In any set of positioning signals, we wish to establish those that traversed direct paths, and use *only* those when ray-tracing.

We can reliably estimate the direct-path subset of measurements by discarding the rays for all measurements that are discarded by the positioning algorithm. In this way we maintain all measurements that are consistent with the returned position. By ensuring that the positioning algorithm is resilient to non-direct measurements, these measurements most likely represent the direct-path subset.

(a) Edge diffraction. (b) Specular reflection.

Fig. 3. Wave phenomena causing signal path deviations.

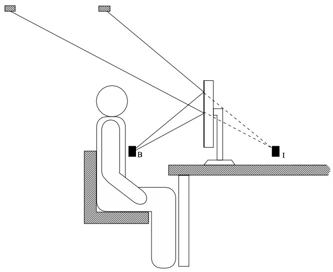


Fig. 4. Reflection images in the Bat system.

However, in extreme cases, it is still possible to misidentify direct signals, as illustrated in Fig. 4. Here, *no* signals propagate to any receivers without first specularly reflecting from a monitor. The resultant time-of-flight (or equivalent) measurements are consistent with a position reflected through the plane of the monitor, and a positioning algorithm will converge on this reflected position, producing rays that obscure the monitor.

The positioning algorithm implemented within the Bat system uses a non-linear model of the data. The standard usage of such an algorithm is to create a model from all measures, discard the largest outlier and repeat until a certain error level is reached, or the algorithm diverges. As described in [6], the Bat algorithm modifies this technique, and discards only the measures with the largest *positive* residuals, thereby encapsulating the idea that an ultrasonic signal cannot travel faster than the speed of sound. The effect of this is that a single direct measurement is enough to prevent the positioning algorithm converging. Thus, situations where a reflected position is returned can be minimised, but not eliminated.

To handle the introduction of such error requires a more detailed approach when forming an occupancy grid. In robotics, a probability of occupancy is assigned to each cell rather than a binary state, and a final probability threshold applied to convert to a binary grid [16]. This works well because the error model for the sensor is easily modelled, allowing for meaningful probabilities. In this methodology, a ray becomes a probabilistic beam with reduced probability of occupancy further from the central axis.

With a positioning system, this methodology can be difficult to apply. Inherent positional error can be used to create a beam, using a Gaussian distribution centred on the position estimate, with a width determined by any available error estimate. However, this does not account for reflections, which are not predictable and hence not reliably modelled. Similarly, if the grid cell size exceeds the typical error for a position, quantising beams onto the grid is little different from the quantisation of rays, but more demanding in both computation and storage.

An alternative approach when the typical positional error is of the same order as, or less than, the cell size is to reduce the effect of erroneous measures statistically. A *voting grid* can be formed, whereby each cell has an associated voting count rather than a binary occupancy state or a floating point probability. This voting count is incremented whenever a ray intercepts the cell.

When reflected positions such as that shown in Fig. 4 are possible, we expect to see a build up of votes within those cells in proportion to the probability of finding the incorrect reflected position. The positioning algorithm described in [6] makes this probability sufficiently small that we can identify the erroneous cell interceptions by thresholding the voting grid to form a binary occupancy grid.

4 Ceiling Mounted Positioning Systems

The systems that have exhibited the highest positioning accuracy to date are primarily ceiling mounted [17, 13, 10, 12]. The Bat system, for example, uses a mobile ultrasonic transmitter and a series of ceiling mounted receivers to position to within 3cm (95% confidence level).

Such systems are well suited to ray-tracing because they typically have a reasonably large density of receivers and produce accurate positions, ensuring a fast and extensive build up of reliable rays. However, the geometry of the situation means that volumes vertically below a surface (such as the space underneath a table) cannot be mapped. So such systems are best restricted to the determination of specific surfaces.

4.1 Horizontal Surfaces

Determining the height, size, and shape of horizontal surfaces in the human environment is particularly useful. These are the surfaces upon which we can work, and upon which we store items of interest to context-aware systems.

To demonstrate ray-tracing, an experiment was performed using a combination of two tables within a room covered by the Bat system. The two tables were chosen to be of differing heights and shapes, and were setup as shown in Fig. 5.

A Bat was moved at random in and around the vicinity of the tables for a period of a few minutes, whilst the Bat system was held in a high update rate to maximise result collection. A total of 6,630 positions were recorded, giving rise to 77,631 rays. The rays were quantised onto a voting grid with dimensions 3.0mx4.0mx2.2m and a cell size of 0.02m. A low threshold was applied, since there were no near-vertical surfaces to reflect positions, and thus very few erroneous rays. Figure 6 shows the resultant ray intersections with horizontal planes at different heights, *slices*, superimposed with the measured outlines of the tables.

(a) (b)

Fig. 5. Table configuration for tests.

At low heights, we observe the intersection distribution to be scarce and highly globular; a result of fewer sightings made at that height. Even so, we immediately see the emergence of the larger square table. The table outline becomes sharper as we approach the table height of 0.7m. Beyond this we observe the circular outline of the second table begin to form. Again, we qualitatively note that the outline is sharpest at a height of 1.1m; the correct height for the second table. We then pass through a region of extensive intersections with no large scale regions present. Above a height of approximately 1.4m, we see small near-circular slices of the individual intersection cones associated with, and centred on, each receiver position (see Fig. 6(t)).

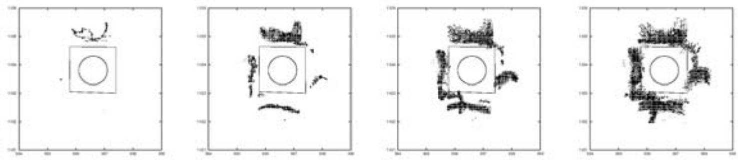
Autonomous Extraction of Specific Surfaces. Once a binary occupancy grid is established, various techniques can be used to extract object details. It is possible to apply image processing techniques such as the Sobel edge detector or the Hough transform [5]. These, however, must be performed in three-dimensional space, which is slow and cumbersome. They also produce extensive edge information, which can be difficult to reduce and amalgamate to form polygonal object representations.

It is also possible to use the characteristics of the occupancy grid at various heights to autonomously extract information about an individual surface. Given a seed point, s, located approximately at the centre of the surface we can use region growing code on a series of slices at different heights to examine the geometrical properties of any region containing s.

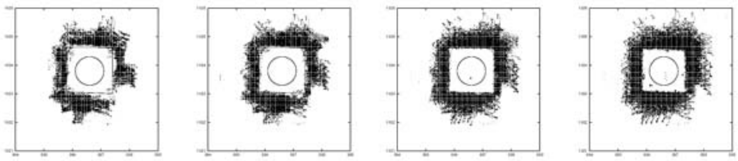
At each height, we can form a dimensionless constant, R, for the region, where R is defined in terms of the region perimeter, P, and the region area, A

$$R = \frac{P^2}{A}. \tag{1}$$

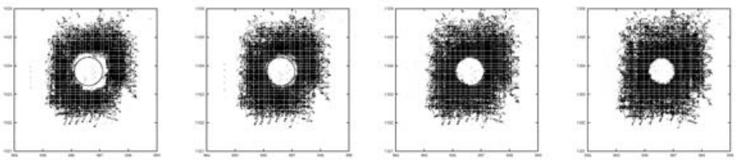
When no region exists around seed s, we stop searching, making the implicit assumption that no surfaces exist above this height. We can justify this by realising that, for there to be no region, rays from the ceiling must penetrate into the area and a significant volume above it, as illustrated in Fig. 7. The dashed horizontal line represents the



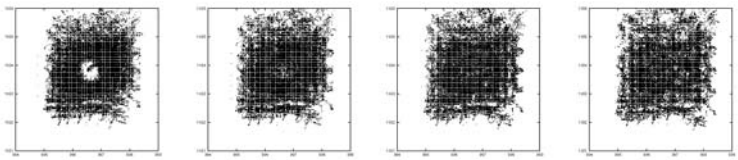
(a) z=0.0m (b) z=0.1m (c) z=0.2m (d) z=0.3m



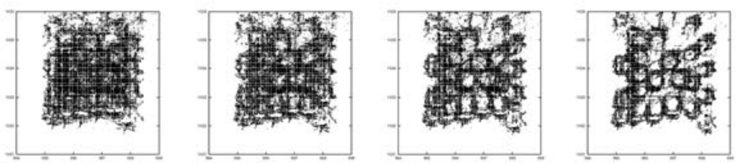
(e) z=0.4m (f) z=0.5m (g) z=0.6m (h) z=0.7m



(i) z=0.8m (j) z=0.9m (k) z=1.0m (l) z=1.1m



(m) z=1.2m (n) z=1.3m (o) z=1.4m (p) z=1.5m



(q) z=1.6m (r) z=1.7m (s) z=1.8m (t) z=1.9m

Fig. 6.

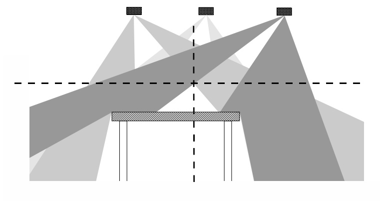


Fig. 7.

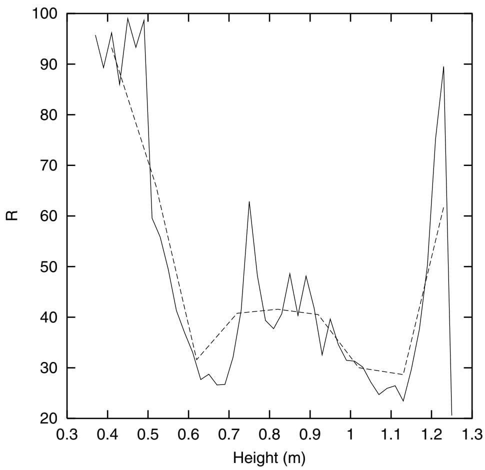


Fig. 8. The profile plot.

first height at which no region can be found using the seed shown as a vertical dashed line. We can conclude that any object above this height would likely be ceiling mounted itself.

The quantity R provides an estimate of how noisy the region shape is. Since we expect edges to be smooth for comfort, safety and aesthetic reasons, we expect to observe a strong local minimum in a plot of R versus height, z. We term such a plot the *profile plot*.

Figure 8 shows the profile plot for the data collected for Section 4.1, using a seed manually calculated to be the centre of the true table position. The underlying distribution is noisy and can be smoothed with a simple boxcar average (dashed line in Fig. 8). This highlights two major minima at heights of 0.7m and 1.1m. We can then find the local minimum within this region from the original profile plot. The results shown in Fig. 8 estimate the heights of the two minima to be 0.67m and 1.07m, in good agreement with the measured heights of the table surfaces (0.683m and 1.090m, respectively).

Figure 9 shows the convex hull of the perimeter of each region extracted at the heights with minima in the profile plot. The true shapes and positions of the tables are superimposed for comparison. Table 1 details the associated errors.

Note that the value of R is also indicative of shape of the object. Table 2 gives the expected values of R for a series of shapes. Whilst it has not been possible to

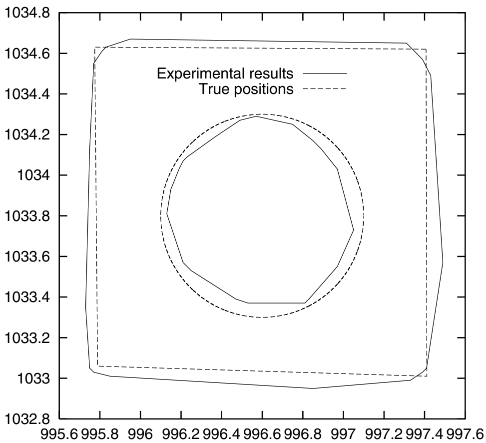
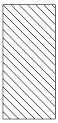
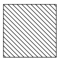
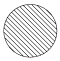
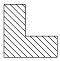


Fig. 9. Autonomous extraction results.

Table 1. Errors.

Table	Perimeter Error	Area Error	**R** Error	Height Error
Square	-0.001m (-0.02%)	+0.28m^2 (+11.2%)	+15.4	-0.013m (-1.9%)
Circular	-0.28m (-0.09%)	-0.15m^2 (-19.5%)	+16.1	-0.02m (-1.8%)

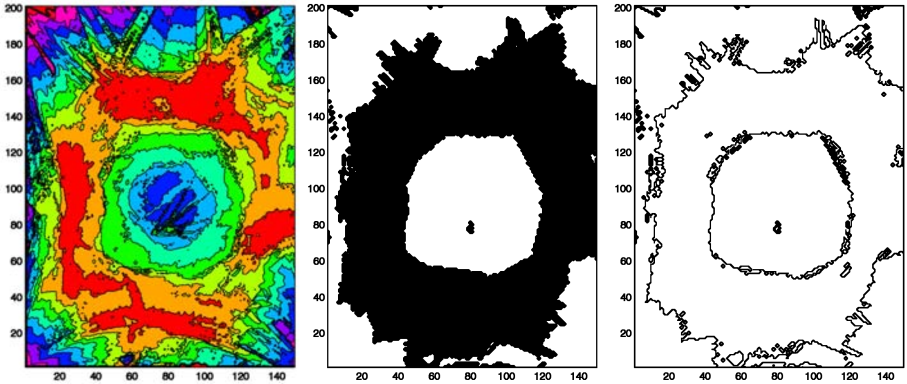
Table 2. Profile plot ratios.

Shape					
R value	18.0	28.8	16.0	4.0	25.0

reliably extract the shape given any experimentally determined value of R, the order of magnitude of R is a useful indicator as to whether an object is genuinely present. For example, finding $R > 50$ is unlikely for an object in an office environment.

Seed Identification. Section 4.1 described a method of extracting a surface shape and height given a seed point which lies within its bounds. To autonomously extract all the surfaces in a region, then, we require a method of identifying the seed points within a data set.

Consider a three dimensional occupancy grid with all rays quantised into it, as described in Section 2. We can examine each vertical column of cells in turn, and create a two-dimensional grid which contains the largest number of consecutive empty cells



(a) Contour plot. (b) Thresholded using average value. (c) Region extraction.

Fig. 10. Using contour plots to determine seeds.

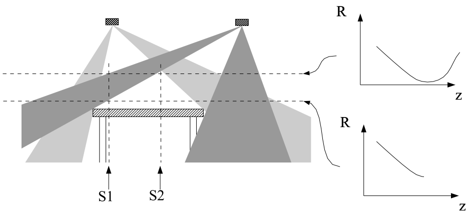


Fig. 11. The importance of seed choice.

within that column. This creates a contour plot of the number of empty cells within a column, as shown in Fig. 10(a) for the table data above.

By applying a threshold to the contour plot (Fig. 10(b)), we can use region growing algorithms to extract regions (Fig. 10(c)) from which we obtain seed points. Figure 11 illustrates that the choice of seed point within a region is important. It shows two seed points, $S1$ and $S2$, and sketches of their resultant profile plots. We observe that $S1$ prematurely ends the plot, and obscures the local minimum. To avoid this situation, we take the centre of the column with the highest count as a seed point ($S2$). This ensures that we do not prematurely cease searching in the profile plot. If multiple cells have this same maximum, it suffices to take the average centre position.

Given a set of seed points, we create a profile plot for each, and search for minima and subsequently horizontal surfaces as in Section 4.1.

4.2 Non-horizontal Surfaces

Non-horizontal surfaces are potentially problematic for ray-tracing in ceiling mounted systems. Finding such surfaces is possible by first performing a coordinate rotation as illustrated in Fig. 12, and then applying the profile plot analysis of the previous section.

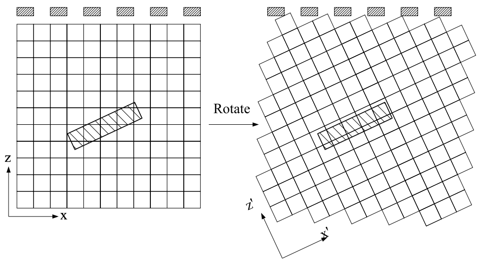


Fig. 12. Coordinate system rotation for non-horizontal surfaces.

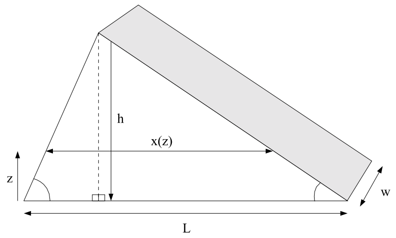


Fig. 13. Wedge example.

Such an approach can work well given prior knowledge of the surface inclination. If this information is not available, as is often the case, we require that a complete profile plot be created for every three-dimensional direction and the most likely plot then determined. This is computationally expensive and hence an impractical solution.

Profile plots may still exhibit useful characteristics. To demonstrate this, a large cardboard wedge was suspended within a room (Fig. 13), and 42,270 rays collected in its vicinity. In effect, the wedge is a stack of horizontal surfaces of diminishing area with increasing height. As expected, then, we see an extended minimum in the profile plot (Fig. 14), which does not lend itself to the analysis of Section 4.1.

However, it is of use to consider the variation of region area with height. A three-dimensional object will have a characteristic trace in a plot of these quantities. For example, consider the arbitrary wedge in Fig. 13. At height, z, its horizontal cross-section has area,

$$A(z) = w \cdot x(z). \tag{2}$$

Where the dimension, x, varies as

$$x(z) = L - \frac{z}{\tan\theta_1} - \frac{z}{\tan\theta_2}. \tag{3}$$

Simple geometry dictates that

$$L = \frac{h}{\tan\theta_1} + \frac{h}{\tan\theta_2}. \tag{4}$$

Combining (2), (3), and (4), then, we find that

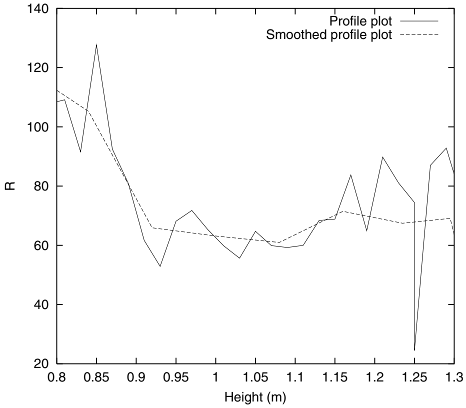


Fig. 14. Wedge experimental profile plot.

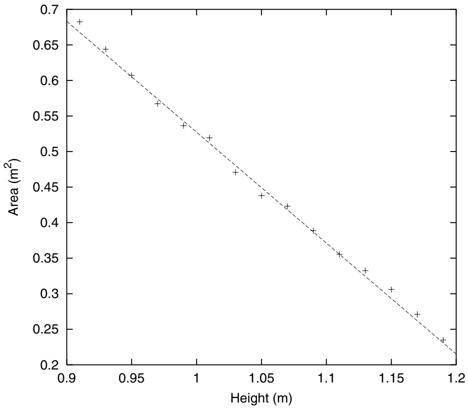
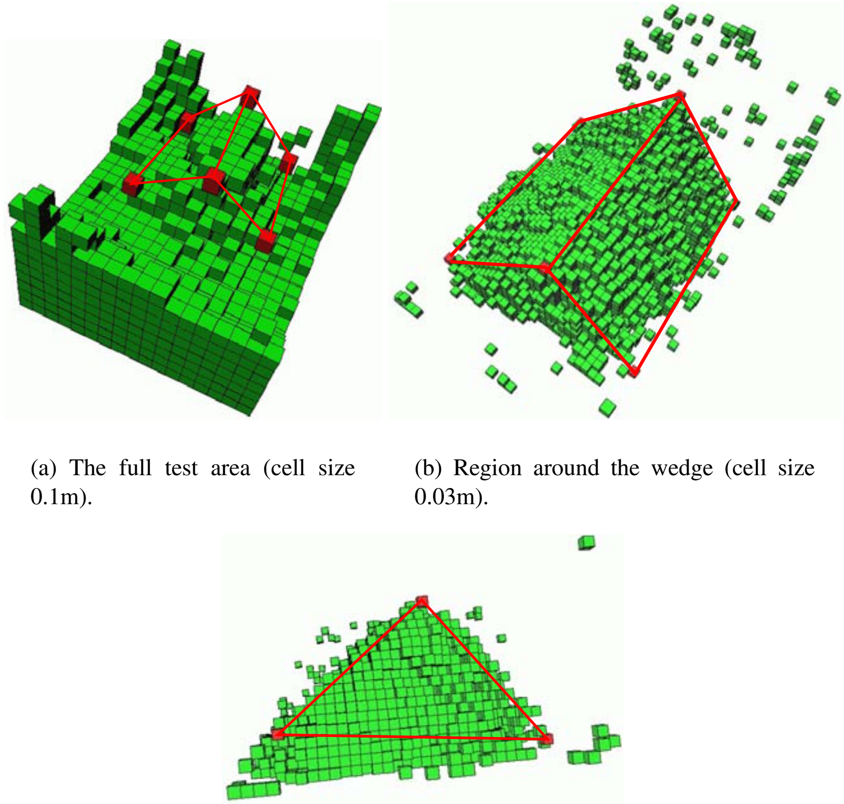


Fig. 15. Variation of area with height.

$$\begin{aligned}\frac{dA}{dz} &= -w \cdot \left(\frac{1}{\tan\theta_1} + \frac{1}{\tan\theta_2}\right) \\ &= -\frac{w \cdot L}{h}.\end{aligned}$$

This relationship is characteristic for a general wedge. Figure 15 shows the plot of area versus height for the experimental data. From (5), and the measured parameters of the wedge, we expect to find a gradient of -1.55m. The best fit line shown in Fig. 15 has a gradient of -1.56 ± 0.01m.

Whilst it is not possible to infer the shape directly from this gradient measurement (due to ambiguities), it serves as an identifier for an object. Such a quantity can be used to determine whether a newly discovered object is truly a new object, or a previously identified object that has moved. Note that more complex objects have a correspondingly more complex variation of cross-sectional area with height.



(a) The full test area (cell size 0.1m).

(b) Region around the wedge (cell size 0.03m).

(c) Region around the wedge (cell size 0.03m).

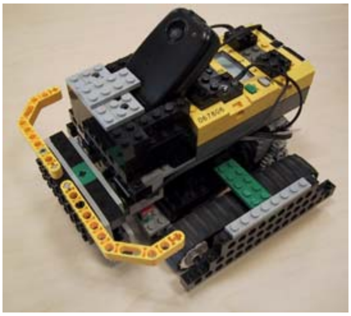
Fig. 16. Unoccupied cells.

In order to analyse new objects of which there is no prior knowledge, volumes can be imaged using the unoccupied cells within an occupancy grid. Figure 16 illustrates the unoccupied cells found in the area of the cardboard wedge. The true wedge shape and position has been overlaid to illustrate the accuracy in determining occupied volume.

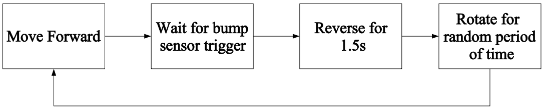
In general, we have found the volume images to represent the top features of objects to a good approximation. Due to the nature of ceiling-based positioning systems, information below these features cannot be extracted.

5 Autonomous Ray Collection

Due to the nature of the Bat system, signals are designed to propagate from Bat height (usually chest height) to the ceiling. This height range does not typically contain any objects, reducing the value of the methods described above when using only personnel movements to collect data. This limitation can be solved by distributing the receivers



(a) Autonomous vehicle.



(b) Movement Algorithm.

Fig. 17.

across a variety of heights, or by using Bats that lie below the height of the objects of interest. The latter approach can be realised with the use of a small autonomous vehicle.

Figure 17(a) shows a simple prototype of such a vehicle, programmed to move with the algorithm of Fig. 17(b). This algorithm effectively moves the vehicle along random paths within an area.

Two tables were arranged within a room as shown in Fig. 18(a). The vehicle, with a Bat attached, was allowed to move freely around the room for a period of 30 minutes. The path taken is superimposed on Fig. 18(a).

The two autonomously identified regions, along with their profile plots, are shown in Fig. 18. The primary minima estimate the table heights to be 0.73m and 0.72m. These correspond to errors of +0.05m and +0.04m, respectively. This is in excellent agreement, giving an error with the same order of magnitude as the underlying positioning system.

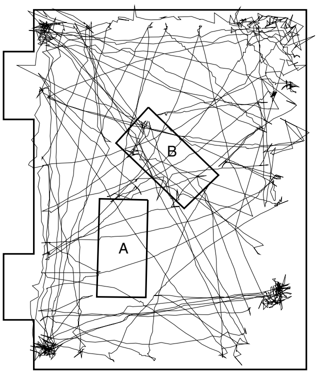
Note that a misleading minimum appears at a height of approximately 0.2m. This is an artifact of all results being collected at this height. Thus the slice at this height is simply the sighting distribution, which is not noisy, and therefore has a low R value. When collecting sightings at a single height, it is hence justifiable to ignore any minimum that is within a small distance of this height.

6 Limitations

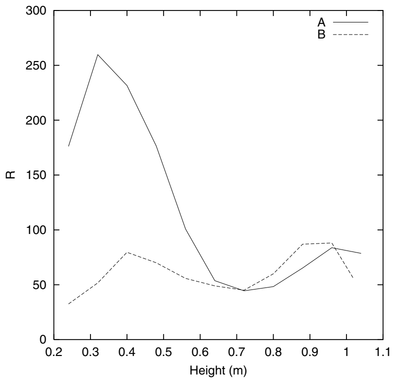
The ray-tracing methodology is limited at any boundary of a ceiling mounted sensor network. In the specific case of the Bat system, ultrasonic signals cannot penetrate walls, making the boundaries rooms.

Positioning itself is limited in the same regions. Consider Fig. 19(a), a topological view of a room and its ceiling mounted receivers. Two transmitter tags, A and B, are shown in symmetrical positions, along with their corresponding areas of possible reception on the ceiling. Since we require a minimum of three measurements to distinct receivers to calculate a three-dimensional position, we see that only B is positioned due to its orientation.

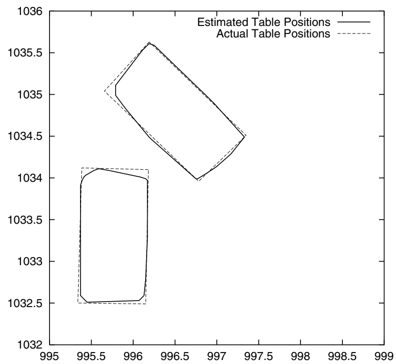
It is possible to rectify the problem by increasing receiver density along the boundaries, but this involves increased installation and maintenance requirements, and does



(a) Setup and path.



(b) Profile plots.



(c) Extraction results.

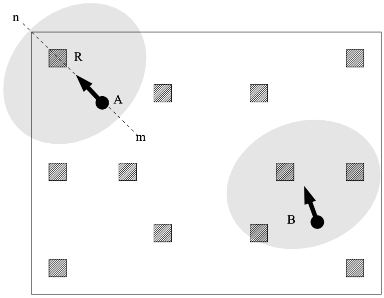
Fig. 18.

not guarantee reliable positioning (since highly clustered receivers do not give a good geometry for positioning). Even when this is done, however, there is a reduced likelihood of transmitters being near to walls.

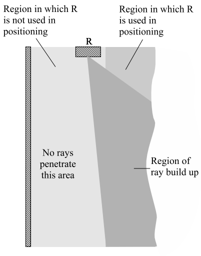
The result is a series of regions near walls and corners where rays do not penetrate because of the asymmetrical receiver distribution, as illustrated in Fig. 19(b). Objects in such regions cannot be found using the ray tracing methods described above. Instead, it is possible to use reflected signals to get information about such areas [6].

It is useful to note that a good receiver geometry for positioning gives rise to a good geometry for ray-tracing, and thus applying the ray-tracing methodology to a positioning system that exhibits good coverage will likely yield good results.

When imaging a volume, the accuracy of the volume shape at representing sloped surfaces depends on the density and coverage extent of receivers, and the actual inclination. Since we typically aim to reduce the density in deployments, and the coverage extent is



(a)



(b) View perpendicular to line n-m.

Fig. 19. Boundary effects.

limited by room bounds, imaging may not extract a clean and accurate representation without a high result density from a diverse set of rays.

7 Conclusions and Further Work

The authors have presented a use of time-of-flight positioning system signals to map the environment. The approach permits for the recognition that an object exists within an environment, and the determination of its height and shape.

The ideas have been implemented and demonstrated to work using results from a ceiling-based ultrasonic positioning system. They should transfer directly to more generic positioning systems that do not have ceiling-mounted receivers. Such systems have fewer limitations, since rays are established in many different orientations, rather than solely toward the ceiling. It is hoped to demonstrate this using a modified Bat system that distributes the receivers throughout three-dimensional space.

This paper primarily addresses the creation of a world model from an unconfigured state. Equally, the process could be used to maintain world models in dynamic environments, although the update rate would be relatively slow, and highly dependent upon the sighting distribution. Future work will address the need for timely and reliable updates to spatial subsections of an occupancy grid.

Acknowledgements

The authors are grateful to both the EPSRC and AT&T for supporting this work through a research grant and CASE award respectively.

References

1. N Adly, P. Steggles, and A. Harter. SPIRIT: a resource database for mobile users. *Proceedings of ACM CHI'97 Workshop on Ubiquitous Computing, Atlanta, Georgia, March*, 1997.

2. Elfes A. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, June 1987.
3. Elfes A. *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1989.
4. I. Getting. The Global Positioning System. *IEEE Spectrum*, 30(12):36–47, December 1993.
5. R. C Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 1992.
6. R. K. Harle, A. Ward, and A. Hopper. A Novel Method for Discovering Reflective Surfaces Using Indoor Positioning Systems. In *Proceedings of the First Int. Conf. on Mobile Systems, Applications, and Services (Mobisys)*, 2003.
7. J. Hightower and G. Borriello. Location sensing techniques. *IEEE Computer*, August 2001.
8. H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9:61–74, 1988.
9. Moravec H. P. and Elfes A. High resolution maps from wide angle sonar. In *Proceedings of the IEEE Int. Conf. Robotics and Autmation*, 1985.
10. N. B. Priyantha and A. Chakraborty and H. Balakrishnan. The Cricket Location-Support System. *Proceedings of the Sixth Annual ACM International Conference on Mobile Computing Networking*, August 2000.
11. D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *Proceedings of GLOBECOM 2001, San Antonio*, November 2001.
12. N. B. Priyantha, K. L. M. Allen, H. Balakrishnan, and S. J. Teller. The cricket compass for context-aware mobile applications. In *Mobile Computing and Networking*, pages 1–14, 2001.
13. C. Randell and H. Muller. Low cost indoor positioning system. In Gregory D. Abowd, editor, *Ubicomp 2001: Ubiquitous Computing*, pages 42–48. Springer-Verlag, September 2001.
14. B. Schilit, N.Adams, and R. Want. Context-aware computing applications. In *Proceedings of the workshop on mobile computing systems and applications*, December 1994.
15. K. Siwiak. Ultra-Wide Band Radio: A new PAN and positioning technology. *IEEE Vehicular Technology Society News*, 49(1), 2002.
16. S. Thrun. Learning maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
17. A. M. R. Ward. *Sensor-driven Computing*. PhD thesis, Cambridge University, August 1998.
18. J. Werb and C. Lanzl. Designing a positioning system for finding things and people. *IEEE Spectrum*, 35(9):71–78, September 1998.

On a Location Model for Fine-Grained Geocast

Frank Dürr and Kurt Rothermel

Institute of Parallel and Distributed Systems (IPVS)
University of Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany
{Frank.Duerr,Kurt.Rothermel}@informatik.uni-stuttgart.de

Abstract. Geographic communication (geocast) is used to send messages to geographic areas, e.g. to distribute warning messages or other information within these areas. It is based on a location model which is used to define a message's target area and the receivers' positions and therefore has strong influence on the achievable granularity of geographic addressing.

A hybrid location model and a fine-grained addressing scheme for geocast based on this model are presented in this paper which support two- and three-dimensional geometric locations as well as symbolic locations like room numbers, embedded local coordinate systems, and mobile target areas like trains.

1 Introduction

The availability of small and mobile devices as well as various positioning systems which can be used to determine the position of these devices enable new forms of communication like geographic communication (geocast). Geocast is used to send messages to certain geographic areas. A typical scenario is the distribution of warning messages. For instance, a geocast message could be sent to all people close to the location where a fire started to ask them to leave this area immediately or just to keep their windows shut because of the toxic smoke. Announcements about traffic jams or car accidents are also interesting mainly for people in the vicinity. An indoor scenario would be the distribution of the slides or other additional information about a talk to all listeners located in the conference room.

Therefore, geocast can be seen as a special kind of multicast. In multicast, a message is sent to a group of receivers. Such a group can be established through explicit join requests by the participants, or it can be established implicitly. In geocast, a group is defined implicitly through the geographic positions of clients and the given geographic target area of the message. The idea is that some messages are only interesting for clients in a certain geographic area. Therefore, all clients inside this target area should receive the message.

As a location-based service, geocast builds on an underlying location model. This location model is used by receivers and the senders of geocast messages:

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 18–35, 2003.

The sender uses the model to define the target area of a geocast message, and the receivers determine whether or not they are in the target area. A client will only deliver a message, if its position is inside the specified target area. Clearly, senders and receivers need a common understanding of locations, which is provided by a location model.

[1] distinguishes between geometric models which define locations by geometric figures and symbolic models using abstract identifiers like street names, room numbers, etc. Most location models used by current geocast implementations like [2] are restricted to two-dimensional geometric models using a global reference system. But for urban areas, it should be possible to use also three-dimensional target areas, defined either by three-dimensional geometric figures or symbolic i dentifiers which are often more intuitive to use, because these are the location identifiers people are used to. Otherwise for example only a whole building could be addressed, even if only the clients inside a small room in this building are to receive a message (see conference example above). Another example for fine-grained geographic addressing using three-dimensional target areas would be a message to all cars on a bridge, e.g. to inform them about an accident or traffic jam, which is irrelevant for the cars passing below the bridge.

Different hybrid models supporting geometric and symbolic locations have already been proposed, e.g. [1, 3]. In this paper, the conceptual design of a hybrid location model for geocast is presented which additionally supports different forms of embedded local geometric and symbolic models enabling e.g. addresses of the form “geometric → symbolic”, i.e. a symbolic location within a geometrically defined area, as well as mobile target areas like ships and trains. The possibly high modelling effort for a fine-grained hybrid location model is reduced by approximation, and model information is used to cope with inaccurate client positions. Although the focus is on a location model for geocast, the presented model may also be applicable to other areas, e.g. to a general location service like [4] that answers so-called range queries (Which objects are in a certain geographic area?).

The rest of this paper is structured as follows. Section 2 presents the system model and gives a short overview of the functionality of the different components used for geocast. Then, the requirements for a location model for fine-grained geocast are stated in Sec. 3. Based on these requirements, a location model is designed in Sec. 4. It will be shown in detail how target areas can be defined by geographic addresses and how they can be compared to client positions based on model information. Finally, an overview of related work is given in Sec. 5, before this paper is concluded with a short summary and an outlook on future work in Sec. 6.

2 System Model

The following components are involved in the delivery of geocast messages.

The *geocast client* is a software component running on a mobile or stationary device like a PDA or PC. It is responsible for sending geocast messages it has received from local *applications* together with the target area address of the message to a geocast router for forwarding. The client is also responsible for delivering messages it has received from a router to applications on its device which are listening for these messages. It knows the position of the device and filters out messages with target areas not containing this client position.

Routers are responsible for forwarding messages from the sender to the clients. According to [2], there are three different ways of geocast message forwarding. **Geographic routing** uses geographic information directly for message forwarding, i.e. special routers compare the target area of the message and the areas covered by sub-networks to decide where to forward the message. The **directory-based approach** stores mappings between geographic locations and IP addresses in a directory and uses this information to determine all receivers in the target area. Messages are sent to each IP address with an associated area which overlaps the target area of the message. The **multicast-based approach** divides the world into so-called partitions with associated multicast addresses. To send a geocast message, first the multicast address of the partition is determined that encloses the target area. Then, a multicast message is sent to this group, that all receivers in this area have joined. A simplified version would be to just broadcast the message and let the receivers filter out messages based on the target area and their current position. Looking at these approaches it is clear that no matter which forwarding approach is used, the target area and the client position always have to be compared before the message is delivered to geocast applications. Where this comparison takes place – at the sender's or the receiver's geocast client, or at some component (e.g. at routers) between sender and receiver – is irrelevant for the conceptual design of the location model, because the model has to provide the necessary information to carry out this comparison in either case. Therefore, the following discussion is independent of the type of forwarding.

Positioning systems are required to determine client positions. Considering their output, two classes can be distinguished [5, 1]: geometric and symbolic systems.

Geometric positioning systems return client positions as geometric figures using coordinates relative to a geometric reference system. This can be a global reference system, e.g. the World Geodetic System 1984 (WGS84) [6] used by the Global Positioning System (GPS), or a local reference system. Especially geometric positioning systems in the indoor domain use local reference systems. Typically, they can only be used in a limited area, i.e. the local coordinate system has a limited scope in contrast to the global scope of for example the WGS84. The Active Bat System [7] – a highly accurate geometric indoor positioning system – for instance uses ultrasound which is blocked by obstacles like walls. If a room is equipped with such a system, only people inside this room can be tracked, i.e. the room is the scope of the system.

Symbolic positioning systems return a symbolic location identifier when a client gets close to a positioning sensor. The Active Badge System [8] for example uses badges transmitting infrared signals, which are picked up by fixed sensors in the environment, so the system can identify the sensors which are within sight of a badge. But identifying nearby sensors is usually only the first step of positioning. Next, the sensor identifiers are mapped by the positioning system onto a symbolic location identifier according to the location model, e.g. a room number, that identifies the client position.

We assume that geometric as well as symbolic positioning systems will be used by clients, e.g. a geometric system like GPS outdoors and a symbolic one like Active Badge indoors. A client possibly knows either a symbolic or a geometric position, but not necessarily both.

Additionally, we assume that the output of positioning systems is not perfectly accurate, i.e. the client position is not defined by a point but by an area called the client area c, e.g. a polygon or the identifier of a room. Therefore, two areas (target and client area) have to be compared, which may overlap partially. The following function is defined that calculates the probability that a client with client area c is in the target area t of a geocast message:

$$p(t, c) \mapsto [0, 1] \tag{1}$$

A value of 1.0 means that the client is completely inside t; 0.0 that it is outside t. If p is greater or equal a specified threshold, then the message will be delivered, otherwise discarded. No matter whether the sender or the receiver defines this threshold, model information can be used to calculate this probability as shown later in Sec. 4.1 and Sec. 4.2.

Finally, the *Spatial Model Service* manages a world model containing location information about countries, cities, buildings, etc. This paper will present in detail which information has to be stored by such a service in order to realize fine-grained geocast.

3 Requirements

Hybrid Model. Because clients can use symbolic as well as geometric positioning systems, a hybrid location model is required to define both kinds of client positions. From a sender's point of view, symbolic locations are often more intuitive to use than geometric coordinates as already mentioned in Sec. 1. Additionally, a symbolic model can be set up with little effort compared to a geometric model, especially for big buildings with many rooms. On the other hand, arbitrary areas defined by geometric coordinates are useful, if there is no or only a coarse-grained predetermined symbolic structure given, which holds especially outdoors, e.g. if some small part of a big plaza is to be addressed. But also if very fine-grained locations are required, a geometric figure might be the better choice. Of course, geometric coordinates should not be limited to two dimensions but three-dimensional figures should be supported as already mentioned in Sec. 1.

Heterogeneous Hybrid Model. We cannot assume that the world is modelled symbolically *and* geometrically at the same level of granularity. For instance, the indoor domain will often be modelled only symbolically, especially if only a symbolic positioning system is available within a building. For other areas, only geometric model information is available. Therefore, a global location model will be built of many partial models, i.e. means are required to integrate local systems into other systems, e.g. a local geometric system into a symbolic one or vice versa. Such a heterogeneous model also leads to situations where the sender specifies the target area of a message geometrically (e.g. by drawing a figure on a map), and only a symbolic client position is known (e.g. if the client is in a building equipped with an Active Badge System) or vice versa. Therefore, a mapping of geometric to symbolic areas or vice versa based on model information is required in order to compare these heterogeneous areas.

Mobile Target Areas. Geocast messages cannot only be sent to static target areas like buildings, but the target area itself may be a mobile object or a location within a mobile object. Typical examples are messages to the dining car of a train or the first deck of a cruise ship. While these objects move, their global coordinates and the coordinates of locations within the object are changing. If these global coordinates were used to specify the target area of a geocast message, this address would also change. Surely, such a constantly changing address is not useful, and thus global coordinates are not appropriate to address these targets. Therefore, means are required to use local coordinates relative to the moving object, which do not change on movement, to address targets within this object.

4 Fine-Grained Location Model

In this section, the design of a hybrid location model is presented. Also an addressing scheme for geocast based on this model is shown and how target areas and client areas can be compared, i.e. how to implement the function specified in Equation 1 that returns the probability that a client is within the target area.

4.1 Symbolic Model

The symbolic part of the presented model consists of a hierarchy of symbolic locations, modelled according to the spatial "contains" relationship. In the example depicted in Fig. 1a, floor F_2 contains the rooms R_3, R_4, and R_5, and building B contains the floors F_1 and F_2, etc. A client located in room R_3 is also located on floor F_2 and in building B. Although this example is rather simple, it illustrates that locations may overlap. For instance, building B contains two wings W_1 and W_2. Wings and floors overlap, e.g. room R_3 is on floor F_2 as well as in wing W_1. In order to be able to model this situation, a model that supports overlapping locations is required. [9] proposed such a model based on the spatial inclusion relationship. That means, for two locations $l_1, l_2 \in L$ where L is the set of symbolic locations, it holds $l_1 \leq l_2$, if l_2 spatially contains l_1. There are two special elements in L named *everywhere* and *nowhere*. The following holds: $\forall l \in L : l \leq everywhere$, and $\forall l \in L : nowhere \leq l$. Mathematically, L and $\leq$

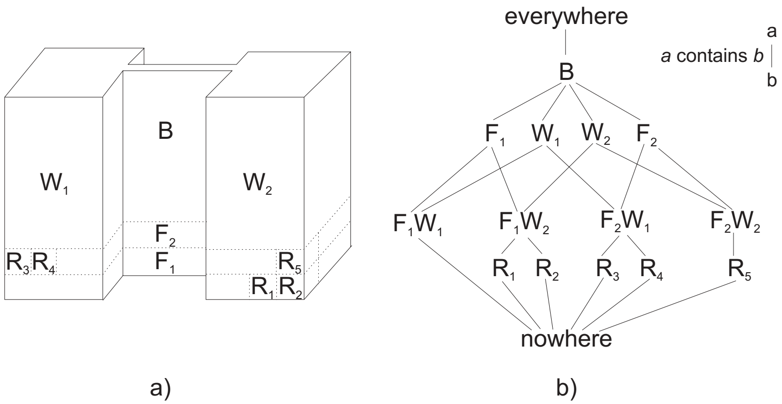


Fig. 1. Hierarchical Symbolic Locations.

form a lattice, i.e. for each pair of elements $x, y \in L$ there is a (unique) least upper bound denoted by $x \sqcup y$ and a (unique) greatest lower bound denoted by $x \sqcap y$. Figure 1b shows the resulting lattice of the example.

Note that this definition of a symbolic location model includes the often used tree-based models, which only allow at most one parent for each location. But a tree is too limited as already shown in the rather simple example above. Neither floors are completely contained in wings, nor wings are completely contained in floors, but each room is part of one floor as well as one wing. A tree cannot model this situation, but the more general and powerful lattice can.

The location hierarchy is used to define symbolic geocast addresses. An address consists of the concatenation of multiple relative identifiers. Each identifier uniquely identifies a location in the context of its parent(s). For instance in the hierarchy shown in Fig. 2, the identifier `room72` identifies room 72 in the context of the location `/de/berlin/keplerstr/9/floor1/wing1`. We use Universal Resource Identifiers (URI) [10] in combination with an XML-based language as syntax for symbolic addresses. The following address stands for the above mentioned location:

```
<targetarea>
 <symbol>loc:/de/berlin/keplerstr/9/floor1/wing1/room72</symbol>
</targetarea>
```

Note that locations may have multiple addresses (e.g. room 72 can also be referred to by the address `/de/berlin/keplerstr/9/wing1/floor1/room72`), whereas each address only stands for exactly one location.

Based on the definition of a symbolic location model, it can now be defined how to calculate the intersection of two symbolic locations. Given a pair of symbolic locations $l_1, l_2 \in L$, l_1 and l_2 overlap, if $l_\sqcap = l_1 \sqcap l_2 \neq \textit{nowhere}$. $l_\sqcap$ is the intersection of l_1 and l_2. This intersection is used in the implementation of the probability function. First the intersection $l_\sqcap$ of the target area t and the

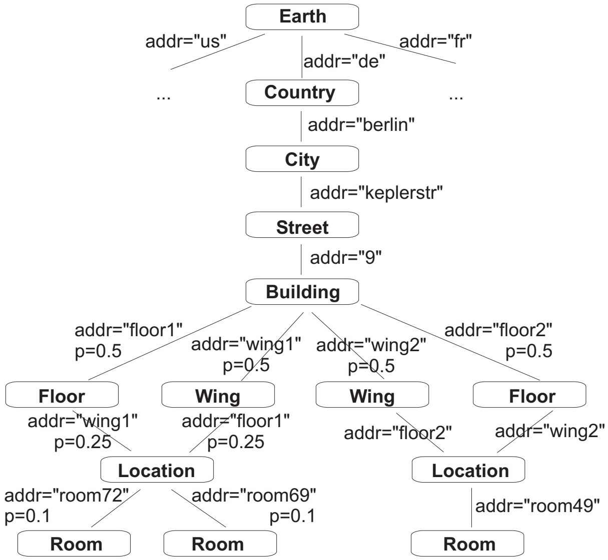


Fig. 2. Symbolic Location Model (some locations like the intersection of floor 1 and wing 2 were omitted for clarity).

client area c is determined by calculating $l_\sqcap = t \sqcap c$. Then, three cases can be distinguished:

- $l_\sqcap = nowhere$: Client area and target area do not overlap at all.
  Example 1: $c =$ `/de/berlin/keplerstr/9/floor1/wing1/room72` and $t =$ `/de/berlin/keplerstr/9/floor1/wing1/room69`, i.e. a message is sent to one room, and the client is located in another room.
- $l_\sqcap = c$: The client is completely within the target area.
  Example 2: $c =$ `/de/berlin/keplerstr/9/floor1/wing1/room72` and $t =$ `/de/berlin/keplerstr/9`, i.e. a message sent to the whole building, and a client in one room of this building.
- $l_\sqcap \neq c$ and $l_\sqcap \neq nowhere$: Target and client area overlap partially.
  Example 3: $c =$ `/de/berlin/keplerstr/9` and $t =$ `/de/berlin/keplerstr/9/floor1/wing1/room72`, i.e. a message addressed to a certain room in a building and the client only knows that it is inside this building, but it does not know the room it is currently in. This may be due to an inaccurate positioning system and a location model that permits target area addresses of a finer granularity than this system can provide (positioning systems used by other clients may be more accurate, and therefore it makes sense to have such a fine-grained model).
  Example 4: $c =$ `/de/berlin/keplerstr/9/floor1` and $t =$ `/de/berlin/keplerstr/9/wing1`, i.e. a message sent to a certain wing of the building and a client that only knows the floor it is currently on. The floor and the addressed wing overlap.

If the client is outside the target area ($l_\sqcap$ = *nowhere*), then p is 0.0; if it is completely within t ($l_\sqcap = c$), then p is 1.0. Partially overlapping locations (case 3) require knowledge about the probability that a client located at c is at the same time at $l_\sqcap$ with $c \geq l_\sqcap$. Therefore, the probability has to be modelled that a client in area c (e.g. building 9) is also located at the sub-locations of c (e.g. floor 1). The following rules hold for a consistent model:

1. If a location is completely partitioned by its sub-locations, then the summed up probability of all disjoint[1] direct sub-locations in a partition is 1.0, because if it is for instance known that a client is in a certain building, then it also must be at some sub-location, e.g. on floor 1.
2. If a location is not completely partitioned (i.e. if for example not all locations within a building have been modelled), then this sum is less than 1.0, because the client may be at an "unknown" sub-location.
3. The summed up probabilities of all disjoint direct sub-locations of a location must never be greater than 1.0, because a client can never be at two disjoint locations of a partition at the same time.

Based on this information, the wanted probability for being at location $l_\sqcap$ can be calculated by multiplying the probabilities on the path from c to $l_\sqcap$. The following example shows how simple rules of thumb can be used to calculate this probability.

Let us assume that building 9 has two floors (and two wings). Then, the probability that a client in building 9 is also on the first floor (or in the first wing) is $\frac{1}{2}$. If there are two floors and two wings, then they overlap at 4 regions, thus the probability of a client on floor $i \in \{1, 2\}$ for being in wing $j \in \{1, 2\}$ at the same time is $\frac{1}{4}$. The rest of the probabilities are given as shown in Fig. 2. Now, the probability of a client in building 9 for being at the same time in room 72 (example 3) is $p = \frac{1}{2} \cdot \frac{1}{4} \cdot \frac{1}{10}$. This estimation can be improved. Room sizes can be used to give larger rooms higher probabilities, or further context information can be taken into account. For instance, the probability that a person is in his own office during working hours might be much greater than for being in another office. In this paper, we will refrain from integrating additional context information.

4.2 Geometric Model

The geometric model contains locations in form of geometric figures, e.g. two-dimensional polygons or circles. The WGS84 is used as global reference system, i.e. coordinates are given by longitude, latitude, and altitude values. These two-dimensional figures are well-suited for objects like countries and cities, but also three-dimensional objects like rooms have to be addressed as mentioned in Sec. 1. But arbitrary three-dimensional figures can be very complex. Thus, not fully-fledged three-dimensional but so-called 2.5-dimensional figures are used. Such a figure is defined by a two-dimensional figure (the base, e.g. a circle or a polygon),

[1] Note that because a lattice is used, sub-locations like floors and wings may overlap.

a fixed height, and the altitude of the base (see Fig. 3). They are only a little bit more complex than two-dimensional figures (the only additional values are the height and altitude of the base), and most common target areas (e.g. rooms, floors, etc.) can be approximated fairly well.

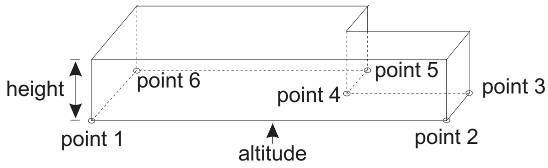


Fig. 3. 2.5-Dimensional Figure.

The geometric figures described above can be used as target area addresses of geocast messages. Here is a simple example using a two-dimensional polygon:

```
<targetarea>
 <polygon><vertex>9.126052E 48.721938N</vertex>...</polygon>
</targetarea>
```

To implement the probability function, first the intersection $c \cap t$ of the client area c and the target area t is calculated by geometric operations (see Fig. 4). Then, the following equation is used to calculate the probability p (note the similarity to the symbolic case: p is calculated based on the degree of overlap of target area and client area):

$$p = \frac{A(c \cap t)}{A(c)} \qquad \text{with } A(x)\text{: area of figure } x \tag{2}$$

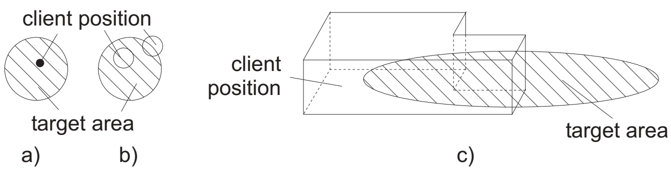


Fig. 4. Comparison of Geometric Client Positions and Target Areas. a) Client position (point) inside target area. b) Client position (two-dimensional figure) overlaps target area (totally and partially). c) Comparison of 2.5-dimensional client position and two-dimensional target area using the base of the 2.5-dimensional figure.

4.3 Hybrid Model

The hybrid location model is formed by associating symbolic locations with their geometric representations, i.e. two- or 2.5-dimensional figures. A perfect model would contain the exact geometry of every symbolic location, but this would mean great modelling effort. A symbolic *and* a geometric client area have to be known as already mentioned in Sec. 3 in order to receive symbolically *and* geometrically addressed geocast messages, because only homogeneous locations can be compared. The following approach does not associate the exact geometry but uses approximation to reduce modelling effort and guarantees that every symbolic location is associated with a (possibly approximated) geometric representation.

Symbolic locations can be *explicitly* associated with their “exact” geometry. We assume that locations high up in the hierarchy tend to have such an explicitly defined geometry. For instance, the outline of most buildings is known or can be determined with little effort from existing plans. At least the borderlines of a city or country are well-known. If a symbolic location is not explicitly associated with a geometric figure, then it inherits the geometry of its parent(s), which in turn may recursively inherit the geometry of their parents, if they have no explicitly associated geometry. If a location has only one parent, then it simply inherits its parent’s geometry. If a location has more than one parent[2], then the intersection of the parents’ associated geometric figures is used, because the lattice-based model defines a sub-location as the intersection of its parents, if it has multiple parents.

Inherited geometric figures are only approximations. They contain the approximated symbolic location completely but they are too big (e.g. a room may inherit the geometry of the whole building, and in the worst case, the geometry of the whole earth is inherited).

When a geometric and a symbolic location – i.e. a geometric target area and a symbolic client area or vice versa – have to be compared in order to determine the probability that a client is inside the target area, the geometric information associated with symbolic locations can be used to translate either the symbolic location to a geometric figure or the geometric figure to a symbolic location. Then, homogeneous locations can be compared as shown in Sec. 4.2 and Sec. 4.1, respectively.

To translate a symbolic location to a geometric figure, the associated geometry of the symbolic location is used.

For the translation of a geometric figure to a symbolic location, there are several approaches. One approach is to associate the smallest symbolic location whose geometric representation encloses the figure entirely. But this may lead to very coarse approximations as shown in Fig. 5. The small geometric area L is translated to the very large symbolic location representing the whole earth, because only the whole earth contains L *entirely.* A second approach is to translate L to multiple smaller symbolic locations, all of which overlap the geometric area at least partially. This results in an approximation that is more accurate. In

[2] Note that this is possible because a lattice is used.

the depicted example, location L could be translated to two symbolic locations representing country A and B. This may result in several necessary comparisons of target area and client position. But in general, only very few symbolic locations will be the result. Therefore, the second approach seems to be the better compromise, if the possibly very coarse approximation of the first approach is regarded.

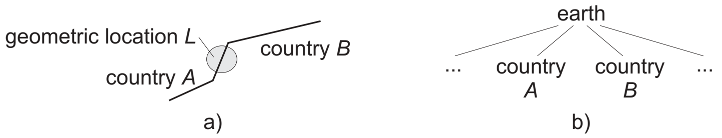


Fig. 5. Translation from Geometric to Symbolic Coordinates a) Small geometric area L overlapping two large symbolic areas. b) Symbolic hierarchy. L can be translated to $\{earth\}$ or $\{A, B\}$.

As already mentioned, the geometry or symbolic location resulting from the translation may only be a more or less accurate approximation. How this approximation affects the delivery of geocast messages is dependent on whether the target area or the client position is translated and therefore possibly approximated.

If the client area is translated, then the translated client area may be bigger than the original area. Thus, according to Equation 2, the calculated probability can be very low, and therefore the message may be discarded depending on the threshold value. Let us for instance assume that the symbolic location $l =$ `/de/berlin/keplerstr/9/floor2/room172` is associated with the (inherited) geometry of the whole building because neither the exact geometry of the room nor the floors are modelled explicitly. Suppose a client located at l, i.e. $c = l$. If a geometrically addressed message is sent to a room of this building and the geometric approximation of l is used in Equation 2, then p will be very low, because the intersection of the target area and the geometric representation of l (the whole building) is very small compared to the geometry of c. But if a geometrically addressed message is sent to the city of Berlin containing this build ing and the room, then the probability will be 1.0 even though the client area has been approximated, because the target area contains the approximated geometry of l completely, and thus the message will be delivered. Therefore, this approximation is only effective for target areas which are bigger than the geometry used as approximation for the symbolic position. This approach should only be used, if it is acceptable that some clients might not deliver a message even though they are in the target area (unimportant messages with high threshold).

If the target area is translated, then an approximated target area is bigger than the original one. Suppose the example shown above and a target area $t = l$. If now this symbolic target area address is translated to its geometric representation, which is the whole building, not only the clients inside the wanted room will deliver the message, but also every client in this building who knows its position geometrically (if it knows its position symbolically, then this position can be compared without translation). Therefore, this approach should be used, if it is acceptable that clients outside the target area deliver the message (important messages with low threshold).

Note that the shown problems related to approximation also hold for the translation of geometric to symbolic locations, because in this case the approximated geometry is used to select one or multiple appropriate symbolic locations as shown above which also may be too big. Therefore, approximation is always only a compromise.

The geometry associated with symbolic locations can also be used to define target areas by combinations of symbolic and geometric attributes. For instance, an address like "200 m around the main station in Berlin" could be defined. Note that this does not require new forms of geographic addresses, because the sender can query the model for the geometric representation of the main station and then uses geometric operations to expand this figure by 200 m. The result is a purely geometric address as described in Sec. 4.2. It has to be mentioned that geometric approximation can make this kind of target address useless in some situations. If for instance a room is approximated by the geometry of a whole building, then it makes no sense to define an area of 5 m around this room, because the approximated geometry is too inaccurate for this short distance.

4.4 Embedded Local Coordinate Reference Systems

Only global coordinates have been considered so far. Now, local symbolic and geometric coordinate systems are introduced which are embedded in global or other local geometric or symbolic systems.

The main difference between global and local systems is the limited scope of the latter, i.e. local coordinates are only valid within this scope. To define the scope of a system, it is associated with a geometric or symbolic location. Each local system has a name that is unique within this area, therefore more than one local system for the same scope can be defined. The scope is used to embed a local system in another higher-level system, and therefore combined addresses of the form "symbolic → geometric", "geometric → symbolic", "geometric → symbolic → geometric", etc. can be specified.

Local coordinates are used in target area specifications by giving the scope and name of the wanted system. The following example uses the system named "sys_room2_72" with a symbolically defined scope (symbolic → geometric):

```
<targetarea>
  <refsys>
   <scope><symbol>loc:/de/.../room72</symbol></scope>
   <name>sys_room2_72</name>
  </refsys>
  <polygon><vertex>2.4 3.0</vertex>... </polygon>
</targetarea>
```

This would be a typical example for an Active Bat System installed in room 2.72, i.e. a local geometric system embedded in a symbolic one. Note that Cartesian coordinates are used for local geometric systems in contrast to geographic global coordinates defined by longitude, latitude, and altitude. If no reference system is given for a geometric figure, then global WGS84 coordinates are assumed.

The next example shows a target area using a local symbolic system embedded in a geometric one (geometric → symbolic).

```
<targetarea>
 <refsys>
   <scope><polygon>...</polygon></scope>
   <name>sys_building9</name>
 </refsys>
 <symbol>loc:floor2/room72</symbol>
</targetarea>
```

The symbolic location `floor2/room72` is interpreted relative to the local symbolic system of building 9. This could be a scenario for a building modelled symbolically, whereas geometric WGS84 coordinates are used outdoors.

Local reference systems can also be nested, i.e. it is for example possible to use a local geometric system embedded in a symbolic system which in turn is embedded in a geometric system (geometric → symbolic → geometric; Fig. 6). This could be necessary, if e.g. a geometric Active Bat System is installed in room 2.72 in the example above, and the symbolic partial model of the building containing room 2.72 is embedded in the geometric WGS84.

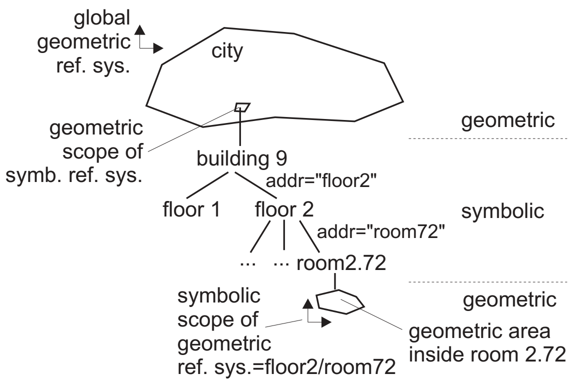


Fig. 6. Multiple Embedded Reference Systems.

Multiple coordinate systems raise the question, what happens if the target area of a message is using system S_1, and a positioning system uses another system $S_2 \neq S_1$, i.e. the client position is known in coordinates relative to S_2? How can these two locations be compared? This question is only of relevance if the scopes of both systems overlap, because otherwise the probability p is 0.0. The answer is that the coordinates of system S_1 have to be translated to system S_2 or vice versa. Three cases have to be distinguished: S_1 and S_2 geometric, S_1 and S_2 symbolic, or S_1 geometric and S_2 symbolic or vice versa.

For two geometric systems, the position and orientation of system S_1 has to be known relative to system S_2 or vice versa. [3] defines the position and orientation of local systems relative to other higher-level geometric systems. Geometric systems are associated with symbolic locations forming a tree. Coordinates are translated by successive coordinate transformations along the path in this tree leading from S_1 to S_2 via other systems. This approach can also be used here.

If S_1 and S_2 are both symbolic systems, then a function mapping symbols from one system to the other is required, if not both systems are using the same symbols. We assume that local symbolic systems do not cover the same area, and therefore, this mapping will not be discussed further in this paper.

If S_1 is a geometric and S_2 a symbolic system, then the procedure described in Sec. 4.3 is used, i.e. the symbolic location in S_2 is translated to its geometric representation in S_1 or vice versa. The scope of the local system can be used as approximation. For instance, the geometric locations within the local geometric system "sys_room2_72" in the first example above can be approximated by the symbolic location – the scope of this system – `/de/berlin/keplerstr/9/floor2/room72`. The symbolic location `floor2/room72` and all other symbolic locations within system "sys_building9" in the second example above can be geometrically approximated by the polygon defining the scope of this system.

4.5 Mobile Target Areas

A last requirement for a location model for geocast is the support of mobile target areas, i.e. moving objects like trains and locations inside such objects.

If the presented addressing concept for global coordinates was used, this would result in constantly changing addresses for locations inside the moving object. Figure 7 shows an example of a train moving from Germany to France[3]. This would also result in a change of the address of the dining car from `/europe/de/ec64/dining-car` to `/europe/fr/ec64/dining-car`. The same problem obviously occurs, if global geometric addresses are used, because the geometric coordinates of the train and therefore its dining car also change during movement.

We also want messages addressed to target areas including the moving object to reach all receivers inside this object. For instance, a message addressed to Germany (`/europe/de`) should also reach the passengers of a train (and its dining car) which is currently located in Germany.

The problem of addressing locations within the moving object can be solved by local coordinate systems as presented in Sec. 4.4. Such a local coordinate system is defined for every moving object, and locations inside these objects are

[3] Locations representing continents have been left out so far for the sake of brevity.

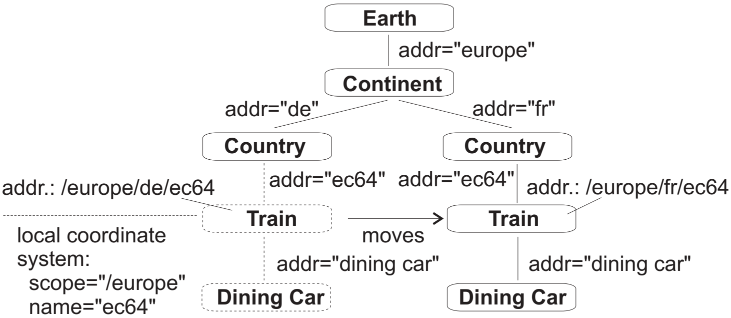


Fig. 7. Mobile Target Area.

addressed by local coordinates relative to this system which do *not* change on movement. The following address specifies the dining car of the train "ec64":

```
<targetarea>
 <refsys>
  <scope><symbol>loc:/europe</symbol></scope>
  <name>ec64</name>
 </refsys>
 <symbol>loc:dining-car</symbol>
</targetarea>
```

Note that the scope of the local coordinate system is static, i.e. it does *not* change when the train moves. Because this is a train only travelling through Europe, the address `/europe` has been used as scope. `dining-car` is the static symbolic coordinate relative to the given local system "ec64". Of course, also local geometric coordinates could be used as shown in Sec. 4.4 assumed a local geometric positioning system is available. To filter received messages, the client's coordinates relative to the system "ec64" have to be known, i.e. a suitable positioning system is required that outputs such local coordinates.

Messages to areas containing the moving object and therefore all clients inside this object do not need special prerequisites as far as addressing is concerned, because global coordinates can be used. To filter globally addressed messages, a global client position has to be known. A global positioning system, e.g. GPS, can be used to determine a client's global position. If such a system is not available – e.g. GPS might not be available inside a railroad car –, the global position of the moving object can be used to approximate a client's global position. Therefore, a location service like [4] is necessary that can be queried for the current location of mobile objects, e.g. the train's position. In our example, the train's position would be `/europe/fr`, if the train is currently located in France. If the train's location is `/europe/fr` and a local positioning system reports that the client is inside this train, then the client's global position is also `/europe/fr`.

5 Related Work

This paper is focused on a location model that fulfils the requirements of fine-grained geographic communication. Most models in this context are rather simple based on geometric locations in a two-dimensional coordinate system. For instance, the geocast implementations of [2] supports simple two-dimensional target areas like polygons and circles in a global coordinate system. Our model goes one step further and also supports three-dimensional (more exactly 2.5-dimensional) and hierarchical symbolic locations. Furthermore, our model supports not only a global but also local reference systems and locations within moving objects. This location model enables users to define destination addresses of finer granularity, and therefore makes geocast applicable in urban areas.

In the context of Geographic Information Systems (GIS), [9] proposes to model spatial relations and operations with lattices. We use this approach as basis for the symbolic part of our hybrid model, extend it with an addressing scheme, and model probabilities for being at certain locations based on the symbolic location hierarchy.

Like the presented model, [11] also uses a lattice-based model. But geometric information is not modelled, and therefore the integration of widely used geometric positioning systems like GPS and also highly accurate geometric indoor positioning systems like the Active Bat System is difficult.

Some approaches from the field of qualitative spatial reasoning and GIS like the RCC-8 [12] and the 9-intersection model [13], respectively, offer a richer set of supported topological relations. For instance, locations "touching" each other can be distinguished from overlapping regions. We think that our hierarchical approach based solely on the spatial part-of relationship is powerful enough to compare target and client areas and still yields models that can be set up easily. Additionally, we introduced probabilities for complex situations where a qualitative statement about the relationship of target and client areas is not sufficient to make decisions about message delivery.

The approach that we consider to be closest related to ours is [3]. This hybrid model also supports symbolic locations as well as geometric representations in local reference systems. In contrast to [3], we do not use a strict tree-based model for symbolic locations but a lattice, i.e. a more general and powerful model that has advantages in more complex situations with overlapping locations. Furthermore, our model supports different forms of embedded local systems, which enables the integration of purely geometrical or purely symbolical partial models. We consider this to be very important for a fine-grained model of global extent that is too complex to be set up by a single authority.

[14] describes another class of location models based on graphs which model locations as nodes and interconnections between these locations, e.g. doors connecting rooms, as edges. Such a model could also be used to define target and client areas *implicitly* by specifying a location and a distance to this location, e.g. everybody within 100 m distance to a certain room. We support similar addresses, but rely on the geometric representation of symbolic locations.

6 Summary and Future Work

The conceptual design of a location model for fine-grained geocast has been presented in this paper. For geocast to be applicable in urban areas, a detailed location model is required to express target areas of geocast messages and client positions. This model has to support geometric locations as well as symbolic locations, embedded local reference systems, and mobile target areas like trains. A formally founded hybrid location model has been designed that supports all these types of locations and the embedding of local geometric and symbolic reference systems, so symbolic and geometric positioning systems can be integrated easily. Additionally, modelling effort is reduced by geometric approximation of symbolic locations. Based on this model, an addressing scheme for geocast has been presented, and it has been shown how homogeneous and heterogeneous locations representing target areas and client positions can be compared by calculating the probability that a client is inside the target area in order to decide, whether a geocast message is to be delivered or not.

The efficient implementation of a service that manages the presented hybrid location model is part of the future work of our project. In [15, 4], we have already proposed a scaleable architecture for the management of geometric location models and a location service that manages large numbers of mobile objects based on geometric location information. To extend this architecture to support also symbolic locations needed for the presented hybrid location model is a big challenge. It has to be scaleable to global extent, which requires a distributed architecture consisting of many servers storing parts of the world model. Geometric *and* symbolic locations have to be managed efficiently to quickly answer queries about spatial containment.

Additionally, a graph-based topological model as shortly described in Sec. 5 will be integrated which enables the implicit definition of areas. Furthermore, such an integrated model can also be used for other tasks like navigation.

The presented location model can be further refined, e.g. by modelling object attributes like class types. This additional information can be used to define groups of receivers within a geographic area, e.g. all taxis near the main station. Messages can then be sent to this group only and not to all receivers in an area.

The design and implementation of suitable geocast protocols is also part of our future work. In [16], we proposed an extension of the geographic routing approach from [2]. Further approaches will be investigated in order to realize fine-grained geocast.

Acknowledgements

The first author gratefully acknowledges the financial support by the Deutsche Forschungsgemeinschaft within the Centre of Excellence 627 “World Models for Mobile Context-Based Systems”. We would like to thank Martin Bauer, Tobias Drosdol, Dominique Dudkowski, Alexander Leonhardi, and the anonymous reviewers for their help to improve this paper, and especially Christian Becker for all the fruitful discussions about location models.

References

1. Leonhardt, U.: Supporting Location-Awareness in Open Distributed Systems. PhD thesis, Imperial College London, Department of Computing (1998)
2. Navas, J.C.: Geographic Routing in a Datagram Internetwork. PhD thesis, Rutgers University, Department of Computer Science (2001)
3. Jiang, C., Steenkiste, P.: A hybrid location model with a computable location identifier for ubiquitous computing. In: Proceedings of the Fourth International Conference on Ubiquitous Computing (UbiComp 2002). LNCS 2498, Göteborg, Sweden, Springer (2002) 246–263
4. Leonhardi, A., Rothermel, K.: Architecture of a large-scale location service. Technical Report TR-2001-01, University of Stuttgart, Institute of Parallel and Distributed Systems (2001)
5. Hightower, J., Borriello, G.: Location systems for ubiquitous computing. Computer **34** (2001) 57–66 IEEE Computer Society Press.
6. Department of Defense: WGS84 Military Standard. `http://164.214.2.59/publications/specs/printed/WGS84/wgs84.html` (1994)
7. Ward, A., Jones, A., Hopper, A.: A new location technique for the active office. IEEE Personal Communications **4** (1997) 42–47
8. Want, R., Hopper, A., Gibbons, J.: The active badge location system. ACM Transactions on Information Systems **10** (1992) 91–102
9. Kainz, W., Egenhofer, M.J., Greasley, I.: Modeling spatial relations and operations with partially ordered sets. International Journal of Geographic Information Systems **7** (1993) 215–229
10. Berners-Lee, T., Fielding, R., Irvine, U., Masinter, L.: Uniform resource identifiers (URI): Generic syntax. IETF RFC 2396 (1998)
11. Brumitt, B., Shafer, S.: Topological world modeling using semantic spaces. In: Proceedings of the Workshop on Location Modeling for Ubiquitous Computing, UbiComp 2001. (2001) 55–62
12. Randell, D.A., Cohn, A.G.: Modelling topological and metrical properties in physical processes. In Brachman, R., Levesque, H., Reiter, R., eds.: Proceedings of the First International Conference on the Principles of Knowledge Representation and Reasoning, Los Altos (1989) 55–66
13. Egenhofer, M.J., Herring, J.R.: Categorizing binary topological relations between regions, lines, and points in geographic databases. Technical report, Department of Surveying Engineering, University of Maine (1990)
14. Bauer, M., Becker, C., Rothermel, K.: Location models from the perspective of context-aware applications and mobile ad hoc networks. In: Proceedings of the Workshop on Location Modeling for Ubiquitous Computing, UbiComp 2001. (2001) 35–40
15. Nicklas, D., Großmann, M., Schwarz, T., Volz, S.: A model-based, open architecture for mobile, spatially aware applications. In Jensen, C.S., Schneider, M., Seeger, B., Tsotras, V.J., eds.: Proceedings of the 7th International Symposium on Spatial and Temporal Databases (SSTD 2001), Redondo Beach, CA, USA (2001)
16. Coschurba, P., Rothermel, K., Dürr, F.: A fine-grained addressing concept for geocast. In: Proceedings of the International Conference on Architecture of Computing Systems 2002 (ARCS '02), Karlsruhe, Germany (2002)

RightSPOT: A Novel Sense of Location for a Smart Personal Object

John Krumm, Gerry Cermak, and Eric Horvitz

Microsoft Research, Microsoft Corporation
One Microsoft Way, Redmond, WA, USA
{jckrumm,gcermak,horvitz}@microsoft.com

Abstract. One of the main prerequisites for location-based services is knowledge of location. We present a simple algorithm for computing the location of a device based on signal strengths from FM radio stations. The motivation for this method comes from a new class of smart personal objects that will receive digital data encoded in regular FM radio broadcasts. Given their built-in ability to receive FM and to measure signal strengths, we show how to exploit this ability to measure the device's location. Our algorithm, called RightSPOT, is designed to be robust to manufacturing variations among devices that affect how they measure signal strength. Toward this end, we present a location classification algorithm based not on absolute signal strengths, but on a ranking of signal strengths from multiple FM radio stations. In tests with three devices in six suburban areas, we show that we can correctly infer the device's location about 80% of the time.

1 Introduction

One of the promises of ubiquitous computing is to connect users to important information as they move around the world. Our research organization has created a small, low-power device platform named Smart Personal Object Technology (SPOT). The first manifestation of a SPOT device will be a commercially available wristwatch, a prototype of which is shown in Figure 1. The SPOT device is designed to listen for digitally encoded data such as news stories, weather forecasts, personal messages, traffic updates, and retail directories transmitted on frequency sidebands leased from commercial FM radio stations. The device holds promise for connecting millions of people to valuable notifications and alerts.

The existing method for localizing data for transmission to particular devices is to depend on the limited range of FM radio signals, so that only devices within range of a particular radio tower will get data relevant to that tower's coverage area. Unfortunately, for certain messages, this coarse location resolution is inadequate. Traffic updates, limited time discount offers, and lists of nearby attractions need finer location filtering than that provided by FM radio station coverage areas. One alternative is GPS, but it does not work indoors, adds volume and expense to an already densely packed device, and consumes precious battery power.

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 36–43, 2003.

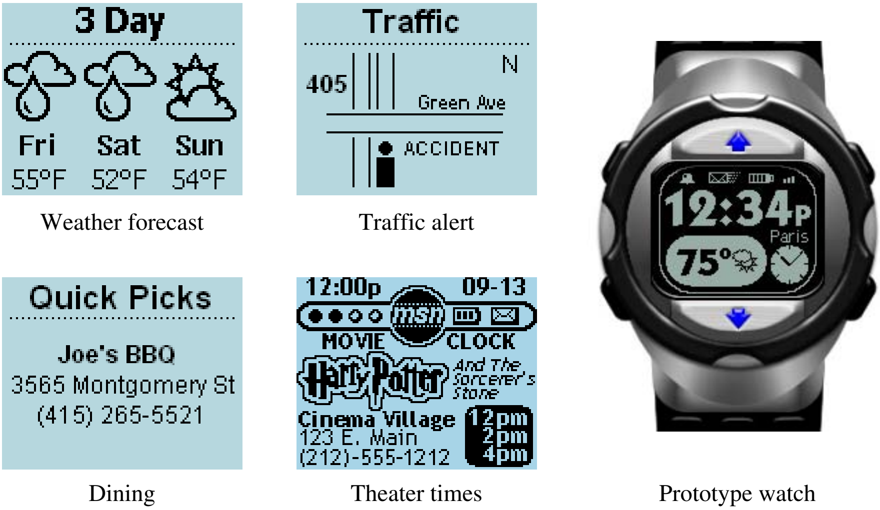


Fig. 1. The SPOT watch displays recent data from updates transmitted over existing, commercial FM radio stations.

Our RightSPOT system embodies a simple method for localizing the device based on signal strengths from existing FM radio stations. The standard SPOT device already contains an FM receiver and the hardware and software necessary to measure signal strength on arbitrary frequencies in the FM band. We show how we can use these signal strengths to localize the device down to a suburb.

The concept of measuring location from radio signal strengths is not new. The RADAR[1] system demonstrated how to localize an 802.11 device based on signal strengths from Wi-Fi access points. Some active badge system, *e.g.* Krumm *et al.*[2], use radio signal strengths for measuring position. Roos *et al.*[3] presented a probabilitiestic framework for determining location from radio signal strength. In our case, the attraction of using commercial FM radio is its wide coverage (indoors and outdoors) and the fact that the SPOT device already has hardware and software on board for measuring FM signal strengths. Despite the fact that ubiquitous, commercial radio broadcasts from fixed towers have existed for a long time, we are unaware of other work aimed at trying to use them to infer location.

2 Measuring Signal Strength

RightSPOT uses a vector of radio signal strengths taken from different frequencies to identify location. Each time a location is to be inferred, the device scans through a set of FM frequencies and records the signal strength of each one.

A standard SPOT device must be able to scan FM radio stations and measure signal strength in order to find a sufficiently powerful one transmitting SPOT data. The “radio signal strength indicator” (rssi) comes from an analogue-to-digital converter (ADC) in the device. The raw digital measurements from each frequency are scaled and then averaged over 20 readings for 13 milliseconds. The ADC and associated circuitry are not carefully calibrated to measure rssi in any certain units nor to be consistent from device to device.

The expected inconsistency among devices for measures of rssi provides a challenge, as such variations complicate attempts to generalize for resuse a single mapping between signal strengths and locations. One solution to this problem is to specially calibrate each device using a source of variable, known FM transmission strengths. The result of one of these tests is shown in Figure 2, showing how the rssi readings of a particular device vary with a known transmitted signal strength. The data for this test was taken in a Faraday cage, and the procedure was deemed too costly for mass production.

Another solution is to train each device after its purchase at different locations, gathering signal strength vectors in known places to be used in the same device later. Regular consumers would likely not tolerate such a training regimen.

In addition to manufacturing variations, signal strengths are also affected by the watch's orientation, its surroundings, and the adjustment of the wrist band which serves as the radio antenna. It would be nearly impossible to anticipate all these variable factors affecting absolute signal strength. If we could anticipate absolute signal strengths, a probabilistic framework like that of Roos *et al.*[3] would be appropriate. However, given the impracticality of discovering each device's response characteristics, we pursued an alternative method of comparing signal strengths.

Rather than depend on absolute signal strength as an indicator of position, RightSPOT employs a *ranking* of a set of radio stations in terms of their measured rssi as described in the next section. While we cannot depend on the devices to give consistent absolute signal strengths, they are designed such that the relationship between input signal strength and output rssi is at least monotonically increasing, such that an

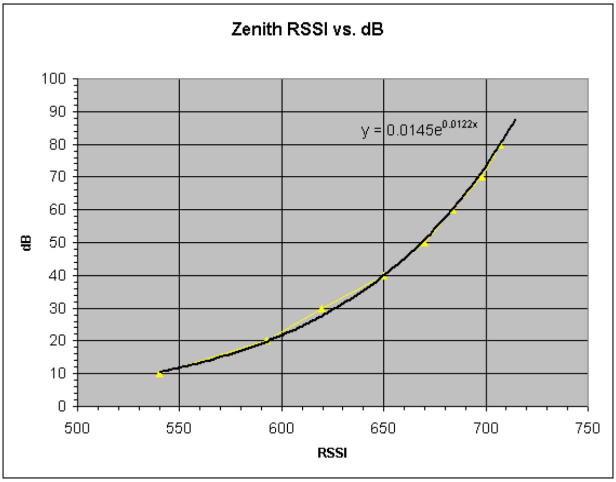


Fig. 2. This is the result of testing one SPOT device to find its transformation between input signal strength and reported rssi. Such an analysis was deemed too tedious to perform on every device.

increase in input signal strength translates to an increase in measured rssi. We leverage this monotonicity in the localization procedure.

3 RightSPOT Location Algorithm

The RightSPOT algorithm infers the location of the device by scanning a list of n FM radio frequencies, $\underline{f} = (f_1, f_2, \ldots, f_n)$ resulting in a corresponding vector of measured signal strengths $\underline{s} = (s_1, s_2, \ldots, s_n)$. In our tests, we varied the number of radio stations n from 2 to 9. A sort routine is used to compute a rank vector of the signal strengths, $\underline{r} = (r_1, r_2, \ldots, r_n)$ in ascending order, where each r_i gives the rank of the corresponding s_i in $\underline{s}$. For example, if the signal strength vector were $\underline{s} = (12,40,38,10)$, the corresponding rank vector would be $\underline{r} = (2,4,3,1)$. We note that the rank vector is insensitive to any monotonically increasing function of the elements of $\underline{s}$, which makes the algorithm robust to variations in how different devices measure signal strength.

For n radio stations, there are $n!$ possible rank vectors, which are the permutations of the integers $1,2,\ldots,n$. Each rank vector can be mapped to an integer $R \in \{0,1,\ldots,n!-1\}$ using a mixed-radix representation of the integers as described by Knuth[4]. Thus we generate a unique hash code for each permutation of signal strengths.

Our classification scheme was motivated by our assumption that different locations will show different relative signal strengths. Ideally each location would map to a single, unique value of R. In reality, due to noise, caused by such factors as the local tilt and position of the SPOT antenna and local electromagnetic effects cause by buildings and terrain, each location produces a distribution of different R's. For training the system, we bring a SPOT device to each of L locations, gathering hash codes $R_i^{(l)}$, where $l = 1,2,\ldots,L$ indexes over the locations and $i = 1,2,\ldots,N_l$ indexes over the hash codes observed at location l. For each location, we can construct a normalized histogram of the hash codes to approximate the discrete probability distribution of hash codes seen at that point, $p(R|l)$. An example of these normalized histograms for six locations and three frequencies is shown in Figure 3.

Given the observation likelihoods $p(R|l)$, and an observation R^*, we can compute the probability of being in any of the L locations using Bayes rule:

$$p(l|R^*) = \frac{p(R^*|l)p(l)}{\sum_{l'=1}^{L} p\left(R^*|l'\right)p(l')} \tag{1}$$

Here $p(l)$ is the *a priori* probability of being at location l. As we have no prior knowledge of the device's location, we assume a uniform distribution, setting $p(l) = 1/L$. Rather than compute likelihoods, we can directly compare nonnormalized

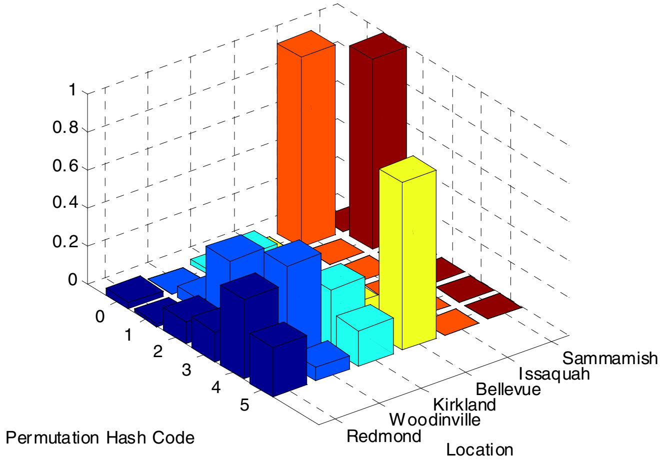


Fig. 3. Probabilities of observing a given ranking (permutation hash code) conditioned on location. Since three radio stations are being measured, there are 3! possible permutations of their signal strengths. This histogram is encouraging, since the "Issaquah" and "Sammamish" locations are almost uniquely identified by permutations 0 and 1, respectively.

posteriors. The Bayes classifier specifies taking the class with the maximum *a posteriori* probability, *i.e.*

$$l^* = \arg\max_{l=1\ldots L} p\left(l \middle| R^*\right) = \arg\max_{l=1\ldots L} p\left(R^* \middle| l\right) \tag{2}$$

Algorithmically, this means that for an observation R^*, we consult the normalized histogram (*e.g*. Figure 3), look up the values of $p\left(R^* \middle| l\right)$ over the full range of locations $l \in \{1,2,\ldots,L\}$, and take the location l with the largest value of $p\left(R^* \middle| l\right)$.

4 Results

To test the RightSPOT algorithm, we chose three SPOT watches at random from our laboratory's store of test devices. We made no effort to choose watches that gave consistent signal strengths, and we made no effort to calibrate the watches with respect to an absolute signal source nor with respect to each other. This simulates the eventual production environment, considering the economic infeasibility of calibrating each device.

We programmed each device to measure signal strengths of 32 different local FM radio stations. In order to download the signal strength in to a PC, the watch had to be mounted in its charger stand, which also provides an RS232 port for data transfer. The RS232 port, in turn, is only active when the charger stand is powered. Thus we put the chargers and watches in the back seat of a car and powered the chargers from

Fig. 4. Shown in boxes are the six suburbs we used for testing our localization algorithm. We took about 10 minutes of signal strength readings from each suburb.

the car's battery. We drove to six different suburbs in our area, logging all 32 signal strengths once per second. In each suburb, we took an average of about 620 readings (~10 minutes) while driving around the suburb's retail core. We chose the retail core as compelling applications of this technology will involve retail businesses sending out time-sensitive offers and listings of local attractions. The six test suburbs are shown in Figure 4.

The raw rssi data from the devices is noisy, as shown in Figure 5, so we applied a windowed median filter to the data, replacing each rssi with the median of itself and the preceding 29 unfiltered values.

We tested by alternately picking data from one of the three devices as the basis for the normalized histograms and testing with data from the other two. This was a more realistic test than merely testing each device against itself, because ultimately all the devices will have to depend on one pre-programmed set of histograms for determining their location.

In an effort to minimize the storage and computational burden of location determination, we also experimented with using a much reduced subset of the 32 recorded radio stations. For each test of n radio stations, we examined the set of $\binom{32}{n}$ different combinations of stations to use for classifying location. The results in terms of classification accuracy are shown in Table 1. The reported accuracy is the fraction of

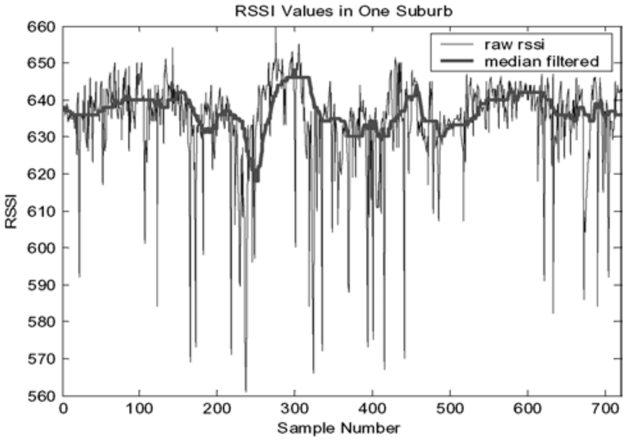


Fig. 5. These are rssi values recorded while driving in the retail core of Redmond, WA. We applied a 30-sample wide median filter to the raw data to reduce noise.

correct inferences made over all the tests where one device was held out for making histograms and the other two used for testing. For $n \leq 5$ we could exhaustively test all possible combinations. For $n > 5$ we tested a random subset of 10,000 combinations. This explains why the classify accuracy goes down when moving from $n = 5$ to $n = 6$ stations: we likely did not find the best combination in the random subset for $n = 6$. The best classification accuracy is 81.7% using $n = 8$ radio stations.

We were pleasantly surprised at how well the simple RightSPOT algorithm could achieve this level of location accuracy using transmitters and receivers that were not intended for providing location information.

5 Conclusions

Our study demonstrates the feasibility of using existing FM radio signals to localize a device down to a suburb using a simple algorithm. For the specific focus of our work on supporting SPOT devices, we leveraged existing hardware to measure FM radio signal strengths, so the capability for localization only requires the addition of a small amount of software to an existing configuration. Different devices measure signal strengths differently, and signal strengths are also affected by many other variables. Our Bayesian classification algorithm does not use absolute signal strengths, but instead uses a ranking of signal strengths to help ensure robustness across devices and other variables. We made our tests realistic by making a random choice of devices, by bypassing calibration, and by not testing the individual devices against themselves.

As we move forward with this research, we will study the question of how to anticipate signal strength characteristics of different locations without actually visiting them. One possible solution is to use a commercially available signal strength simulator like RadioSoft's ComStudy software. Such a simulation could also help determine

Table 1. Classification accuracy generally increases by including more radio stations in the analysis.

Number of Radio Stations = n	Best Subset of FM Radio Stations	Classification Accuracy
2	{KPLU, KEXP}	34.3%
3	{KWFJ, KEXP, KVTI}	61.6%
4	{KWFJ, KEXP, KSER, KVTI}	72.9%
5	{KEXP, KSER, KVTI, KBCS, KLSY}	77.6%
6	{KVTI, KGHP, KRWM, KLSY, KEXP, KSER}	76.2%
7	{KBCS, KEXP, KUBE, KBSG, KVTI, KSER, KLSY}	79.7%
8	{KVTI, KSER, KBCS, KJR, KNHC, KEXP, KBSG, KUBE}	81.7%
9	{KVTI, KRXY, KPLU, KJR, KBCS, KLSY, KUBE, KSER, KEXP}	76.8%

a good subset of radio stations to listen to for best localization. Another interesting problem is how to determine which radio station histograms to transmit to a device, especially if we can make no *a priori* assumptions about where in the world it might be. Finally, we might realize a boost in classification accuracy if we were to smooth position inferences over time, adhering to constraints about how fast devices are expected to move between locations.

References

1. Bahl, P. and V.N. Padmanabhan. RADAR: An In-Building RF-Based Location and Tracking System. in IEEE INFOCOM 2000. 2000. Tel-Aviv, Israel.
2. Krumm, J., L. Williams, and G. Smith. SmartMoveX on a Graph -- An Inexpensive Active Badge Tracker. in Ubicomp 2002. 2002. Goteborg, Sweden: Springer.
3. Roos, T., P. Myllymaki, and H. Tirri, A Statistical Modeling Approach to Location Estimation. IEEE Transactions on Mobile Computing, 2002. **1**(1): p. 59-69.
4. Knuth, D., Seminumerical Algorithms, in The Art of Computer Programming. 1981, Addison-Wesley: Reading, Massachusetts. p. 64 (Algorithm P).

User-Friendly Surveying Techniques for Location-Aware Systems

James Scott[1] and Mike Hazas[2,⋆]

[1] Intel Research Cambridge
[2] Lancaster University
james.w.scott@intel.com

Abstract. Many location-aware applications rely on data from fine-grained location systems. During deployment such systems require a *survey*, specifying the locations of their environment-based components. Most current surveying methods are time-consuming, and require costly and bulky equipment.

This paper presents the concept of *self-surveying*, i.e. methods by which a location system can survey itself. Such methods are user-friendly, fast, and require little or no extra equipment. Experimental results show self-survey accuracies comparable to the accuracy of the underlying location system.

1 Introduction

Ubiquitous computing applications often make use of location information. There has been a recent focus on *fine-grained* location systems, which are capable of accuracies down to several centimetres. Unfortunately, existing location systems, particularly those that are fine-grained, typically require much effort and expertise to deploy. This presents a significant obstacle to getting location systems (and consequently ubiquitous applications) out of research labs and into homes and offices. While it is foreseeable that miniaturisation, power management, and wireless communication will facilitate user-friendly *installation* of location system infrastructure, the requirement for easy *surveying* of location systems has not been widely discussed.

With regard to location systems, a survey is a collection of data concerning the physical configuration of the system infrastructure. Such infrastructure often comprises ceiling– or wall-mounted components, and a survey might include accurate positions and/or orientations of these components. In the fine-grained location systems developed and installed in research labs, the surveying procedures used often rely on specialised methods and equipment, which may not be suitable for wide deployment scenarios. Development of simple, inexpensive, and user-friendly surveying methods is therefore important.

This paper discusses the concept of *self-surveying*, which is the ability of a location system to perform a survey of itself, using its own infrastructure and components.

⋆ The experiments discussed in this paper were conducted while both authors were members of the Laboratory for Communication Engineering (LCE), University of Cambridge.

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 44–53, 2003.

2 Related Work

This section presents some conventional surveying methods, and describes prior work in self-surveying. First, however, some fine-grained location systems are briefly reviewed.

In the *Constellation* [1] system, a mobile unit containing ultrasonic sensors is worn on the head and belt of the user, and transmitters are placed in the environment. By using the times-of-flight of ultrasonic pulses, the user is tracked with accuracies of about 5 mm. The *Bat* [2] system employs ultrasonic ceiling receivers to track small transmitting tags worn by people and attached to objects, with accuracies of about 3 cm. Adopting a more privacy-oriented approach, *Cricket* [3] uses independent ultrasonic transmitters in the environment and mobile receivers, with reported accuracies between 5 and 25 cm. Additionally, recent research in ultra-wideband radio systems indicates their potential for accurate tracking [4].

The *EasyLiving Person Tracker* [5] uses two stereo vision cameras to track up to three people in a room. However, this system is not able to directly identify the subjects. The *TRIP* system [6] uses a single camera to detect and identify circular barcode tags worn by people and attached to objects. Provided a tag is within 3 m of a camera, its position can be estimated with an accuracy of 10 cm. *HiBall* [7] is a location system designed for augmented and virtual reality. Users each wear a small device consisting of a cluster of lateral effect photo diodes. Arrays of ceiling infrared LEDs are flashed in sequence, allowing the position of each wearable to be estimated with typical accuracies of about 5 mm.

2.1 Conventional Surveying

For many fine-grained location systems described in the literature, there has been no significant discussion concerning the surveying procedures used. In practice, either the environmental units are mounted in a deterministic fashion so that their relative positions are known [3], or units are arbitrarily placed and then surveyed using manual measurements [8].

Another class of surveying techniques uses physical devices to aid the surveying process. Two such devices are the *crate* and *theodolite*. The crate, developed at AT&T Labs Cambridge, is a purpose-built device comprising a rigid frame with three spring-loaded reels of cable mounted on it; each reel can electronically report the length of its extended cable. By touching the end of each cable to each survey point, trilateration software can determine the points' locations. A theodolite is a device which accurately measures the range and horizontal and vertical angles between itself and a reflector. This tool, normally used by land surveyors, was used at the LCE to survey a Bat system deployment.

To assess the accuracy of the crate and theodolite, Bat system infrastructure in three rooms was surveyed using both devices. Distances between forty ceiling unit pairs were measured by hand using a tape measure. The difference between the hand-measured distances and the distances according to the surveys averaged 4 mm for both devices. While both surveying methods are accurate, they are time-consuming (a typical room took one person-hour to survey) and the devices are expensive and cumbersome, and not suitable for user-executed deployments in areas such as homes.

2.2 Previous Research on Self-surveying

There has been some research pertaining to self-surveying with particular fine-grained location systems. Foxlin et al. [1] proposed an *auto-mapping* algorithm whereby three seed beacons are placed at measured locations and other beacons are placed arbitrarily. As a user moves through the environment the tracking system can slowly discover and position the arbitrarily-placed beacons. With regard to ultrasonic location systems, Ward commented that it would be possible to use a non-linear multilateration algorithm to estimate unknown ceiling receiver positions by gathering a number of transmitter-to-receiver distances from transmitting tags placed at known locations [8, page 59].

Gottschalk et al. proposed and implemented an *auto-calibration* method for the HiBall system [9]. Using rough LED location estimates, and thousands of observations from unknown test locations, the system (1) estimates the wearable's position at each test location, and (2) calculates back-estimates of LED positions. These two steps are repeated until LED position estimates converge. In more recent work with HiBall, Kalman filters have been used to accomplish on-line auto-calibration [7]. The methods allow the system to continually update LED location estimates during normal operation.

The work described above is concerned with specific location systems. In contrast, this paper discusses general principles of self-surveying, and describes self-surveying concepts and techniques which are applicable to many types of location system.

3 Self-surveying

Most fine-grained location systems operate by using the surveyed locations of *fixed units* in the environment in order to calculate the location of *mobile units*, such as people or equipment. The survey may include information on either or both of the fixed units' positions and orientations. For example, the vision-based TRIP system relies on knowing the orientation of the fixed units as well as their positions, whereas the ultrasonic Cricket system relies only on knowing the positions.

In normal operation, location systems use sighting data and known fixed unit locations to calculate a mobile unit's location; i.e. they use a number of known quantities to determine a few unknown quantities. However, sightings may contain more data than is strictly necessary for location purposes. For example, in the Bat system, a minimum of three distances would be necessary to locate a mobile unit, but a given ultrasonic pulse is often "heard" by well over three receivers. This *surplus data* is what makes self-surveying viable. Additional surplus data can be generated by constraining the locations of mobile units during self-surveying, thus reducing the number of unknowns, and allowing more data to be put toward finding the fixed unit locations.

To conduct a self-survey, a system must first *gather* sightings which include surplus data. This data must then be *processed* in some way to determine estimates of the fixed unit locations. In addition, there is often a requirement for *combining* data so that self-surveys of nearby physical spaces can be unified into a single coordinate space.

3.1 Data-Gathering Methods

Three data-gathering methods, known as the *people*, *floor* and *frame* methods, were investigated in this research, and are described below.

People. A convenient data gathering method is to log sightings from a location system while it is in ordinary use. Sightings can be gathered as people (equipped with mobile units if necessary) move around the space. This method potentially provides a completely transparent surveying process, whereby a new system would automatically be surveyed as it is used, with survey accuracy improving as the system ages. However, the people method provides less surplus data compared to the methods below, since the locations of the mobile units are completely unconstrained.

Floor. A second self-surveying method would be to place many mobile units on the floor of a space for a period of time. Unlike in the people method, where each sighting may be taken with the mobile unit at a different location, in the floor method, many sightings are gathered from each stationary mobile unit. This makes the self-survey solution set much more constrained (since the mobile units are approximately coplanar), and also provides the advantage that erroneous sightings can be more easily identified. This method requires a dedicated self-survey measurement phase before the system can be put into use, and the user is required to place and recover the mobile units.

Frame. In the frame method, mobile units are placed onto known points on a rigid frame. The frame method's advantage is that it provides the most surplus data of the three methods presented. Since the relative locations of the mobile units are known, the self-surveying algorithm need only solve for the locations of the fixed environmental units. Although this method incurs the disadvantage of requiring hardware for the frame, the frame might be manufactured using inexpensive, lightweight materials for easy assembly.

3.2 Processing Self-survey Data

Once data gathering is complete, the data must be processed to find estimates of the fixed unit locations. Two procedures for accomplishing this are discussed below.

Simulated Annealing. One possible method for finding the fixed unit locations is to use an iterative algorithm to find better and better solutions for the mobile and fixed unit locations until a "best guess" is found. For each possible solution, there must be some method of *scoring* the solution against the gathered data. For example, in the case of an ultrasonic location system, a suitable scoring function might be found by taking the sum of the differences between the distances reported by the sightings and the corresponding distances according to the current location estimates.

Simulated annealing [10] is a mathematical optimisation technique which mimics the process of crystallisation in bulk matter through a randomised search procedure, using the notion of *temperature* to avoid being trapped by local minima in the solution space. Simulated annealing was chosen for the purposes of this research because it involves no assumptions about the solution space; it can thus be applied with little modification to all three data-gathering methods.

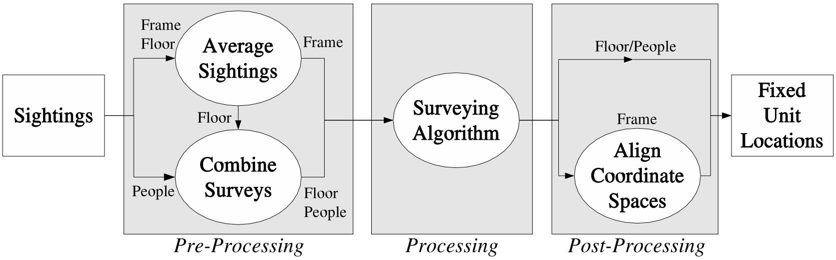


Fig. 1. Data flow for self-surveying

Inverting the Location Algorithm. Since the relative mobile unit locations are known when the frame method is used, it is as if the location system were "inverted," in that a fixed unit must be located using known locations for mobile units. For some location systems, it may be possible to determine the locations of the fixed units by employing the same algorithm that is used for normal operation of the system.

3.3 Unification of Coordinate Spaces

One issue that must be dealt with in many surveying procedures is combining the results of multiple surveys with arbitrary coordinate spaces (such as separate rooms) into a single, unified space (such as a building). This is especially significant for a self-surveying system, as the aims of minimising the work required by the users for surveying would be compromised if they were required to manually measure the relative positioning of the various spaces surveyed, so that coordinate spaces could be combined.

For the people and floor methods, automatic unification of coordinate spaces can be achieved by combining the self-surveying data in a pre-processing stage; i.e. by merging multiple data sets so that they may be presented as a single data set to the processing algorithm. For the frame method, combining data from different surveys during pre-processing is not viable in all cases, since the processing method might rely on knowing the relative locations of the mobile units. Therefore, a post-processing combination of surveys must be performed. This may be achieved by calculating coordinate space transforms which map the points in multiple surveys such that the sum of distances between common points is minimised in the unified survey.

The flow of data during the self-surveying process, and the differences in this flow depending on the method used, are illustrated in Fig. 1.

4 Implementation

In order to determine the accuracy of self-surveying, the Bat system installation at the LCE was used to conduct self-surveys in a number of rooms of various sizes. In particular, two large offices (5 m $\times$ 6 m) and three small offices (5 m $\times$ 3 m) were used for data gathering; however, only two of these five data sets were used as test data during development of the self-survey algorithms.

4.1 Data Gathered

The data was gathered by capturing raw sighting information from the Bat system. The logging included all distances recorded between the surveying tags and the fixed ceiling units during the course of a survey.

For the people method, data was gathered by having two people, each with four tags, walk around a room for a specified period of time. The tags were worn on the chest and on the back, and one was carried in each hand, the hands being kept a forearm's length away from the body. Readings were taken for a period of ten minutes for a small room, and twenty minutes for a large room.

For the floor method, in small rooms twenty tags were placed randomly but uniformly over the floor area, and left for ten minutes. Large rooms were surveyed as if they were two adjacent small rooms, i.e. in two ten-minute runs each covering opposite halves of the room. The data for large rooms was combined at the pre-processing stage so that it appeared as a single forty-tag experiment.

For the frame method, a frame measuring 1 m $\times$ 1 m was equipped with twenty-one tags. In small rooms, the frame was placed in the centre of the room for ten minutes. Large rooms were surveyed by treating them as two adjacent small rooms; the survey results were combined during a post-processing stage, as described in Sect. 3.3.

4.2 Pre-processing

The floor and frame methods use a pre-processing stage to obtain an averaged value for each tag/fixed unit distance. This is accomplished using the following procedure: (1) ignore tag and fixed unit pairs which returned fewer than t distances; (2) for each remaining pair, find groups of distances which make up at least proportion p of the number of sightings between that pair, and which are within d of one another; (3) take the mean of the group representing the shortest distance (since longer peaks are likely to be due to reflections). By trial and error, suitable values for the parameters t, d and p were found to be 10, 7.5 cm, and 40% respectively.

4.3 Processing

Both the simulated annealing and inverted processing techniques were implemented. For the simulated annealing algorithm, the starting location of the fixed units is a single location 2 m above the level of the tags. The tags also start at a single location, except when using the frame method, where the tag locations are specified according to the frame geometry. Ten thousand iterations are used for the floor and frame methods, and twenty thousand are used for the people method.

During each iteration, movements are potentially made to the position of each fixed unit and each tag, except that for the frame method, the tag positions are fixed, and for the floor method, the height of the tags is fixed. Each potential movement is chosen from a group of scalar multiples of a random vector; the scalars used are the integers from -4 to 4. For each of the eight possible new positions, errors in the new tag–fixed unit distances are summed; this provides the scoring function. If at least one of the potential movements gives a better score than the current position, the best-scoring movement is taken. If not,

Table 1. Per-room self-survey accuracy for various data-gathering and processing methods. Accuracies represent the mean location error of all fixed units in the room. Rooms marked * were used as test data when developing the self-surveying procedure. All values are in centimetres

Method	Small Room 1	Small Room 2	Small Room 3*	Large Room 1	Large Room 2*	Mean accuracy	Mean accuracy (central units)
People	20	20	24	17	13	19	15
Floor	12	21	26	9	6	15	5
Frame (SA)	9	9	2	10	5	7	4
Frame (NR)	3	3	2	4	3	3	2

then with probability equal to the temperature, a randomly chosen movement will be taken. The starting temperature is 0.02 for the people method and 0.01 for the frame and floor methods. The temperature declines linearly, reaching zero at the final iteration. At each iteration, if the average distance error is the lowest so far recorded, then the fixed unit locations are recorded.

The above steps are repeated twice, with a *culling* stage in the middle. This stage was introduced to handle erroneous sightings which represent reflections or other errors. While the algorithm largely ignores such errors, the results are still influenced slightly. The culling stage examines each tag's distances, and discards distances which have a much higher error than the others. A second set of iterations is then run.

For the inverted processing technique, the non-linear regression algorithm used by the Bat system was applied. As described in Sect. 3.2, only the frame data may be used with this method. To distinguish between the frame-based results sets, the names *frame (SA)* for simulated annealing and *frame (NR)* for non-linear regression will be used.

5 Experimental Results

For each self-survey, the resulting locations were compared with a theodolite survey. The self-survey accuracies for each room are shown in Table 1. Since simulated annealing produces slightly different results each time it is run, the people, floor, and frame (SA) accuracies given represent the ninetieth percentile.

The results show that the frame (NR) method performs best out of the self-surveying methods, with a survey mean error of 3 cm. Significantly, the large-room experiments exhibited similar accuracies to the small-room experiments, indicating that unification of coordinate spaces during the pre– and post-processing stages did not adversely affect the accuracies of the self-surveying methods studied.

5.1 Analysis and Optimisation

In order to analyse the results further, it is useful to examine an error distribution for individual fixed unit positions, as shown in Fig. 2. The diagram illustrates an "elbow" in the frame (SA) and floor results; while 75% of the fixed units are located accurately, the remaining 25% are for some reason subject to high errors. This is explained by looking at a plan view of one of the rooms, shown in Fig. 3, which shows that large

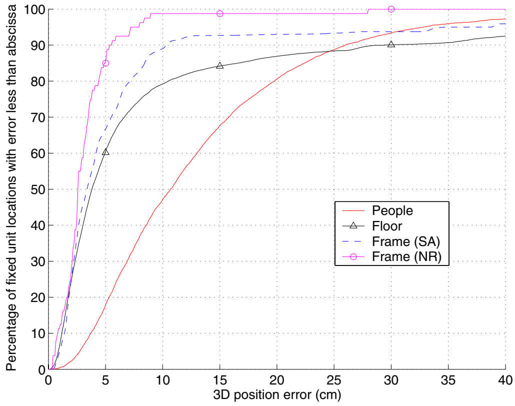


Fig. 2. Location error distributions for all fixed units in all rooms

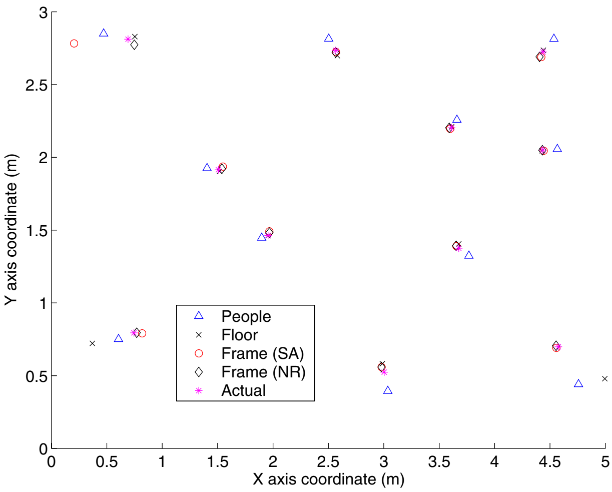


Fig. 3. Plan view of typical self-survey results for a small room

errors occur most often for fixed units located at room extremities. If such "edge units" are removed from consideration, the average floor and frame (SA) room accuracies are greatly improved, as shown in the rightmost column of Table 1.

The reason behind these errors lies in the fact that, especially for the floor and frame methods, the location solutions for edge units will tend to be less rigidly constrained by the data. Such units are likely to have a less favourable geometry with respect to the surveying tags; i.e. they have a poorer position dilution of precision. Also, edge units tend to be at further distances from the surveying tags, and it is known for the Bat system that ranging at distances greater than several metres succeeds less often, and is less accurate. A remedy to both of these problems may be to gather more surveying data over a wider area.

5.2 Accuracy Limits

The results, both for the frame (NR) case and for the central units in the frame (SA) and floor cases, show accuracies comparable to the Bat system's accuracy of 3 cm. This is a very positive result for self-surveying, as it indicates that the accuracy of self-surveying techniques can approach the accuracy of the surveyed system. Note that it is not claimed that a self-surveyed location system will locate objects as accurately as a conventionally-surveyed location system, as the error in self-surveying and error in the location process may combine, either additively or otherwise. Characterisation of this effect is an area for future exploration.

The implication of these results for the deployment of location systems is that there is no single correct surveying method to use. For applications with low accuracy requirements, a minimal-cost, people-based survey may be appropriate. As accuracy requirements increase (e.g. with the deployment of a new application), a more accurate and more labour-intensive survey could be carried out.

6 Further Work

Some further improvements to specific aspects of the self-surveying procedure have already been proposed. This section looks at broader research directions.

One area for exploration is in different types of processing methods. In particular, the non-linear regression algorithm may be extended to be applicable to the floor and people methods. Alternatively, other processing algorithms, for example based on Kalman filters, may be a fruitful area for investigation.

Another area for future work is in validating the viability of self-surveying in other location systems. The Bat system used in the experiments above employs distances from transmitting tags to receiving ceiling units to determine location. It is easy to see how the same methods would be applicable to other ultrasound-based systems, including systems with roaming receivers and fixed transmitters such as Cricket, and to systems in general which utilise time-of-flight methods, such as ultra-wideband systems.

Vision-based systems can also make use of distance. However, they may instead rely on the angle from the camera's axis to the target, or its relative orientation. The viability of self-surveying in such cases may be seen by noting that vision-based systems can operate with roaming tags and stationary cameras, or vice versa. As with time-of-flight systems, it is therefore possible to conceive that the locations of the fixed units can be regarded as unknown values, and that a self-survey can take place, given enough "surplus data."

Finally, moving away from self-surveying, there are other issues in user-friendly deployment of location systems to be explored. One such issue is that of finding the physical configuration of a space outfitted with a location system. This includes finding the locations of walls and doors, of desks and other furniture, and of interactive objects such as computers and telephones. Without this information, location-aware applications cannot assist in users' interactions with their environment. Further work is therefore required in this area; one interesting thread of research involves the use of ray-tracing to model horizontal surfaces in the environment [11].

7 Conclusions

This paper has discussed the concept of *self-surveying* for indoor fine-grained 3D location systems, and has described several self-surveying methods. Experimental results using the Bat system have shown that self-surveying is capable of reaching survey accuracies similar to that of the underlying location system, in this case 3 cm.

Self-surveying facilitates the deployment of location systems by untrained personnel, with minimal overhead in time and hardware costs. By contrast, conventional surveying techniques have required trained use of high-cost and bulky equipment. Self-surveying is therefore a key step toward user-friendly deployment of location systems, and enabling the use of ubiquitous computing applications in everyday environments.

Acknowledgements

The authors would like to thank Rob Harle for his help with the Bat system and insightful comments on this work, Alastair Beresford and Alastair Tse for their work on theodolite surveying, the reviewers for their helpful comments, and Andy Hopper for his support.

References

1. Foxlin, E., Harrington, M., Pfeifer, G.: Constellation: A wide-range wireless motion-tracking system for augmented reality and virtual set applications. In: Proceedings of SIGGRAPH '98, Orlando, Florida, USA (1998) 371–378
2. Addlesee, M., Curwen, R., Hodges, S., Newman, J., Steggles, P., Ward, A., Hopper, A.: Implementing a sentient computing system. IEEE Computer **34** (2001) 50–56
3. Priyantha, N.B., Miu, A.K.L., Balakrishnan, H., Teller, S.: The Cricket Compass for context-aware mobile applications. In: Proceedings of MobiCom 2001, Rome, ACM/IEEE (2001)
4. Lee, J.Y., Scholtz, R.A.: Ranging in a dense multipath environment using an UWB radio link. IEEE Journal on Selected Areas in Communications **20** (2002) 1677–1683
5. Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., Shafer, S.: Multi-camera multi-person tracking for EasyLiving. In: Proceedings of the Third IEEE International Workshop on Visual Surveillance, Dublin, Ireland (2000)
6. López de Ipiña, D., Mendonça, P., Hopper, A.: TRIP: A low-cost vision-based location system for ubiquitous computing. Personal and Ubiquitous Computing **6** (2002) 206–219
7. Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K., Colucci, D.: The HiBall tracker: High-performance wide-area tracking for virtual and augmented environments. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, London (1999)
8. Ward, A.: Sensor-Driven Computing. PhD dissertation, University of Cambridge (1998)
9. Gottschalk, S., Hughes, J.F.: Autocalibration for virtual environments tracking hardware. In: Proceedings of SIGGRAPH '93, Anaheim, California, USA, ACM (1993) 65–72
10. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220** (1983) 671–680
11. Harle, R., Hopper, A.: Building world models by ray-tracing within ceiling-mounted positioning systems. In: Proceedings of UbiComp 2003, Seattle, USA, Springer-Verlag (2003)

Sto(ry)chastics: A Bayesian Network Architecture for User Modeling and Computational Storytelling for Interactive Spaces

Flavia Sparacino

flavia@media.mit.edu

Abstract. This paper presents sto(ry)chastics, a user-centered approach for computational storytelling for real-time sensor-driven multimedia audiovisual stories, such as those that are triggered by the body in motion in a sensor-instrumented interactive narrative space. With sto(ry)chastics the coarse and noisy sensor inputs are coupled to digital media outputs via a user model, which is estimated probabilistically by a Bayesian network. To illustrate sto(ry)chastics, this paper describes the museum wearable, a device which delivers an audiovisual narration interactively in time and space to the visitor as a function of the estimated visitor type. The wearable relies on a custom-designed long-range infrared location-identification sensor to gather information on where and how long the visitor stops in the museum galleries and uses this information as input to, or observations of, a (dynamic) Bayesian network. The network has been tested and validated on observed visitor tracking data by parameter learning using the Expectation Maximization (EM) algorithm, and by performance analysis of the model with the learned parameters.

1 Introduction

Computation and sensing are moving from computers and devices into the environment itself. The space around us is instrumented with sensors and displays, and it tends to reflect a diffused need to combine together the information space with our physical space. Museums and new entertainers are looking into technology as a partner to help them engage their public by generalizing the communication model of the theme park into that of the culture or information park. These interactive spaces need to be able to articulate a story triggered by people with natural interaction modalities, such as those determined by the body in motion in the space, and by gestural or voice input.

Interactive space design involves three elements: human authors (space designers) who conceive flexible and updatable strategies to deliver appropriate information to the public interactively; computers and sensors that gather and process information in real time about the public's behavior in the instrumented space; and the audience, with its own needs to receive personalized content only when and where it is appropriate. Telling stories in interactive spaces requires new flexible sensor-driven and user-centered authoring tools, designed with these three elements in mind.

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 54–72, 2003.

Namely, traditional interactive storytelling techniques have been conceived for non sensor-triggered environments, such as the desktop computer, and rely on mouse click and key presses as their main input modality. They have been authored with techniques that can be categorized to be either scripted, or responsive, or occasionally, behavior-based [1]. All of these systems imply a fixed modality of user-story interaction and story authoring. The premise of this research is that to communicate effectively a sensor-instrumented interactive narrative space needs to be authored with more sophisticated techniques than the ones above. It needs to have the depth of content of a scripted system, the flexibility of a responsive system, and the autonomous decentralized architecture of a behavioral system. It also needs to go beyond the behavioral scheme and respond not just by weighing stimuli and internal goals of its characters, but also by understanding the user's intentions in context, and by learning from the user.

Sto(ry)chastics, is a first step in the direction of having suitable authoring techniques for sensor-driven interactive narrative spaces. As illustrated in the following sections, sto(ry)chastics allows the interactive experience designer to have flexible story models, decomposed in atomic or elementary units, which can be recombined into meaningful sequences at need in the course of interaction. It models both the noise intrinsic in interpreting the user's intentions as well as the noise intrinsic in telling a story. We as humans do not tell the same story in the same way all the time, and we naturally tend to adapt and modify our stories to the age/interest/role of the listener. This research shows that, Bayesian networks are a powerful mathematical tool to model noisy sensors, noisy interpretation of intention, and noisy stories.

2 Related Work

Sto(ry)chastics is related to work in wearable computing, user modeling, and Bayesian networks.

Oliver, Shiele, and Jebara [2] developed a wearable computer with a visual input as a visual memory aid for a variety of tasks, including medical, training, or education. This system records small chunks of video of a curator describing a work of art, and associates them with triggering objects. When the objects are seen again at a later moment, the video is played back. The museum wearable differs from the previous application in many ways. DYPERS is a personal annotation device, and as opposed to the museum wearable, it does not attempt to perform either user modeling or a more sophisticated form of content selection and authoring. The museum wearable in contrast focuses on estimating the visitor's type and interest profile to deliver a flexible user-tailored narrative experience from audio/video clips that have been prerecorded. These clips or animations would usually be part of the museum's digital media collection. Feiner [3] has built a university campus information system, worn as a wearable computer. This device is endowed with a variety of sensors for head tracking and image registration. Both the size of the wearable, mounted on a large and heavy backpack, as well as the size of the display, are inappropriate for it to be used for a museum visit. The device also does not have a computation storytelling engine or a content selection mechanism which adapts to different user types.

The work of Pearl [4] is fundamental to the field of Bayesian networks. Jordan's book [5] had the merit of grouping together some of the major advancements since Pearl's 1988 book. Pynadath and Wellman [6], use a Bayesian network approach to induce the plan of a driver from observation of vehicle movements. Starting from a model of how the driver generates plans, they use highway information as context that allows the system to correctly interpret the driver's behavior. While applied to traffic monitoring, and not to sensor-driven, user-centered interactive storytelling their research bears the most resemblance to the one here presented. Albrecht et al [7], have been amongst the first to model the behavior of a participant in a computer game using Bayesian networks. They use several network structures to predict the user's current goal, next action, and next location in a multi-user Dungeon adventure game. With respect to their work sto(ry)chastics performs not just user state estimation, which can also be used to predict the next action that the user will do, but it also adds a probabilistic mechanism for content selection. This research also bears resemblance to Maes' work on intelligent agents who assist users in daily tasks such as buying and selling stocks or finding music recommendations based on the user's ratings of a given music database [8]. The focus of sto(ry)chastics is to function as a sensor-driven computational system that potentially uses the same unifying mathematical framework (Bayesian Networks) for sensor modeling (tracking, gesture recognition), user estimation, and content selection.

Sto(ry)chastics' content selection process is inspired by work in probabilistic knowledge representation. Koller and Pfeffer [9] have done innovative work in using probabilistic inference techniques that allows most of the frame based knowledge representation systems available today to annotate their knowledge bases with probabilistic information, and to use that information to answer probabilistic queries. Their work is relevant to describe and organize content in any database system so that it can later be selected either by a typed probabilistic query or by a sensor driven query. Using a content database, annotated probabilistically, sto(ry)chastics selects the most appropriate content segment at each time step, and it delivers, interactively in time and space, an audiovisual narration to the museum visitor as a function of the estimated visitor type.

3 The Museum Wearable

Rather than describing sto(ry)chastics in general, it is easier to focus on a specific application. As an example of application of sto(ry)chastics, and to illustrate its features, I designed a real time storytelling device: a museum guide which in real time evaluates the visitor's preferences by observing his/her path and length of stops along the museum's exhibit space, and selects content from a set of available movie clips, audio, and animations. In this document I ground on this specific application, called the Museum Wearable, further discussion on modeling first the user's type, then estimating his/her interest profile, and subsequently the selection of content. For more details on the museum wearable device [Fig. 1], its construction, software architecture and potential impact on exhibit design please see [10]. Additional references to the wearable literature, and comparison with hand-held devices can also be found in this paper. Here the reader will also find a more thorough discussion on the museum literature that has led the author to identify the busy, greedy, and selective types as valid classes of museum visitors to design personalized content for. The specific focus of this paper is on the Bayesian network that estimates the user model, and the mathematical technique used to ground this model on visitor tracking data gathered at the museum.

Fig. 1. The Museum Wearable

4 Estimating the Visitor's Type with a Bayesian Network

The museum wearable uses a Bayesian network to provide a real time estimate of visitor types: a greedy type, who wants to know and see as much as possible, and does not have a time constraint, a busy type who just wants to get an overview of the principal items in the exhibit, and see little of everything, and the selective type, who wants to see and know in depth only about a few preferred items [10]. Following are the assumptions used to model the Bayesian network that estimates the visitor's type:

1. The information available to the system, modeled by the observed nodes of the network is location (which object the visitor is close by) and how long the visitor stays at each location (duration).

2. The priors on the three busy/greedy/selective types start with equal values for the three types. The visitor does not declare belonging to a user type at the entrance so as not to bias the priors on the visitor type to any of the three types. Some visitors might in fact feel bothered being asked or being committed to a type at start.

3. The visitor type is represented by a static node rather than a dynamic node. It can change but it is not a variable that evolves in time. What this means is that even if

the visitor starts with a busy behavior and later behaves selectively, the system can still account for that while considering the visitor as a unit and not as an entity whose value we sample in time. This is to a certain extent a subjective choice of the modeler (the author), guided by commonsense and previous modeling experience.

4. To initially simplify the task, I have selected a subset of twelve representative objects at the MIT Museum's Robots and Beyond Exhibit [Fig. 2].

Once the basic modeling assumptions have been made various choices are still available when designing or modeling a Bayesian network in practice. It is clear that the model needs to be some kind of dynamic Bayesian network, as estimating the visitor's type is a process that happens in time during the visit. Sampling times are given in this case by the presence of the visitor at each location, and therefore we need to model a process with twelve time slices, as twelve are the selected objects for the exhibit. In addition to this, the actual placement of these objects on the museum floor dictates some constraints on the model architecture. Therefore the geography of the objects in the exhibit needs to be reflected in the topology of the modeling network [Fig. 2]. The choices available in modeling this problem are:

- the topology of the network
- which states per node
- whether to have a continuous or discrete node for the visitor's stop durations
- whether to have one unique node or multiple nodes to describe the stop durations.

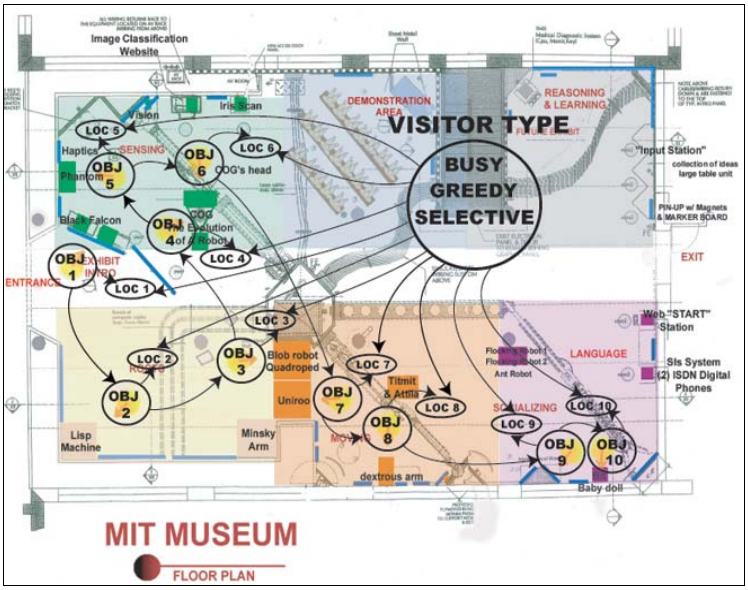


Fig. 2. Bayesian Network topology superimposed on the MIT Museum Robots and Beyond exhibit's map

One possibility is to consider a model which has the topology of an HMM. However such a model would require having a dynamic visitor node, which is a choice I discarded in the above discussion on the working hypothesis for this model. A similar topology, but with a static visitor node, can be obtained by introducing an object node which encodes information about the object being observed by the visitor. This is shown in Fig. 3 for the first three objects/time slices.

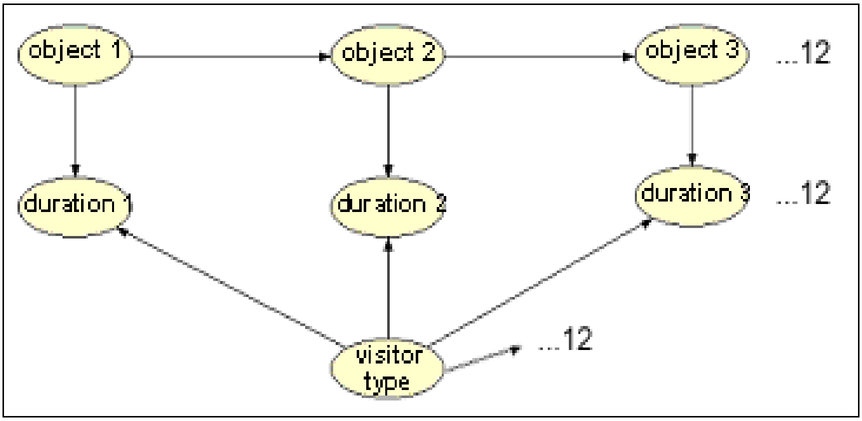


Fig. 3. Bayesian Network used to estimate the visitor type

For example, some objects can be very interesting or less interesting, according either to the opinion of the curator, or the public's preferences, and this information can be encoded in the Bayesian network. Moreover, it would be possible, theoretically, at the end of each week, to take the posterior probabilities for all objects and all visitors, and reintroduce them as priors of the Bayesian network the following week. This would allow the system to account for the evolving public's preferences and types. While installation of the museum wearable in the museum is not addressed in the research described by this document, it is important to notice the possibility to easily update the priors of the model from the posterior probabilities of the previous day or week to obtain a more accurate estimate of the variables of interest.

The Bayesian network has three nodes per time slice: the object and location nodes are dynamic, i.e. they are repeated at each time slice, whereas the visitor node is a static node, as I assume that the visitor type does not change during the visit at the museum. The input to the network are the location nodes, which express how much time the visitors spend with each object. Rather than having location nodes with continuous values, which would require more sophisticated probability update techniques, it is more convenient to have discrete location nodes with a few states, such as "short", "long" stop duration, or "skip" object. Section 5 describes how, mathematically, stop durations are labeled as skip/short/long. It is of course possible later on to extend the model to consider a higher resolution of discrete stop durations, such as one which includes: 'skip', 'short', 'medium', 'long', and 'very long'. These ranges are specific to each museum exhibit and are derived from tracking data obtained by observing and measuring visitor's stop length and type at the museum.

Classifying people's behavior using the proposed dynamic Bayesian network can be done on the sole basis of the expert's opinion, which assigns numerical values to the conditional probability tables of the network. This results in an expert system. In some cases performing probability update on such models leads to satisfying results, such as for systems that model the specialist's medical knowledge in decision support systems which help determine whether further tests or investigation are needed to assess the health state of a patient. In the specific case of this research, the expert's initial assumptions about the percentage of skip/short/long stop durations that a busy/greedy/selective visitor will do are given in Table 1. This table assigns numeric values to the qualitative description of these three types. These values are derived from interviews with the MIT Museums' staff in charge of the Robots and Beyond exhibition.

Table 1. Conditional Probability Table for the visitor node assigned by the expert

Conditional probability table for the visitor node			
	% skip	% short	% long
Busy	0.2	0.7	0.1
Greedy	0.1	0.1	0.8
Selective	0.4	0.2	0.4

However, when a database of cases is available, we have the opportunity to perform out-of-sample testing to validate that the model has any predictive power, given the data. This is important in the case of a museum exhibit in which the public's behavior cannot necessarily be correctly modeled by the curator's expertise, because of the changing nature of the shown artwork, and the public's needs. According to the problem that is defined there are various techniques for data analysis which combine prior knowledge with data to produce improved knowledge. In some cases we may need to learn the topology of the network, in others, the topology is given and we need to learn the parameters of the network. Specifically for this research, I made the assumption that the structure of the network is correct and I gathered tracking data on visitor's path and length of stops at the museum to learn the conditional probability tables (parameters) for the nodes of the network.

During the course of several days a team of people at the MIT Museum tracked and make annotations about the visitors. Each member of the tracking team had a map and a stop watch. Their task was to draw on the map the path of individual visitors, and annotate the locations at which visitors stopped, the object they were observing, and how long they would stop for. In addition to the tracking information, the team of evaluators was asked to assign a label to the overall behavior of the visitor, according to the three visitor categories described above. Together with the curator, and in accordance to the indications of the Visitor Studies Association (VSA, http://museum.cl.msu.edu/vsa/whoweare.htm), I have found that allowing the evaluators to make a subjective judgment about the visitor's behavior is as accurate as asking the visitors themselves. In addition to that, the museum wearable acts as an external observer, which tailors a personalized story to the visitor, on the basis of external observations, as opposed to asking the visitor what they want to do at every step.

Lastly, the assessment made by the team of evaluators is used to initialize the Bayesian network, while the model can later be refined, that is the parameters can be fine tuned as more visitors experience the exhibit with the museum wearable, as described later in this document. We tracked 65 visitors, discarded 15 incomplete or invalid annotations and retained 50 valid tracking sheets. Some examples of tracking data are summarized in Table 2. The table contains raw data, that is the number of seconds that visitors stayed in front of the corresponding objects. All these objects were visited in a linear sequence, one after the next, with no repetitions. We observed a total of 3 greedy types, 16 busy types, and 31 selective types.

Table 2. Example of visitor tracking data for the first 10 visitors. Data shown is in seconds

Intro 1	Lisp 2	Minsky Arm 3	Robo Arm 4	Falcon 5	Phantom 6	Cogs Head 7	Quad 8	Uniroo 9	Dext Arm 10	Kismet 11	Baby Doll 12	TYPE
0	5	5	0	13	0	10	0	0	0	0	0	busy
0	0	20	0	30	40	0	0	30	5	24	0	slctv
0	0	0	0	10	0	75	0	0	0	0	10	slctv
0	0	20	0	20	130	10	55	82	25	0	5	slctv
0	0	15	10	10	5	0	0	0	0	0	0	busy
0	0	5	5	5	0	3	0	0	70	0	0	busy
0	0	33	0	60	17	0	0	0	16	0	13	slctv
0	38	10	13	38	10	21	0	0	18	0	43	slctv
0	0	30	0	0	10	10	0	0	5	0	0	busy
0	6	40	15	25	40	0	82	82	34	30	18	Greedy

4.1 Model Description

To fully define the proposed dynamic Bayesian network we need to assign the prior probabilities of the parent nodes and the probabilities of the child nodes conditioned on the state of their parents, that is assign or learn the conditional probability tables. We also need a transition probability table between one time slice and the next.

I initially implemented the network, for simplicity, with a binary object node with visited/not-visited states. The reason is that for the training of the model I later performed, and which is discussed in the next section, no priors were available on whether the objects were more interesting than neutral or boring. I would have had to learn separate conditional probabilities for the neutral/interesting/boring cases, for which no training data was available. After having taken tracking data on the twelve selected objects, it is possible to analyze this data and infer priors on whether the objects are neutral/interesting/boring. Then we need to give different conditional probabilities for the interesting and boring cases, such that if a busy type spends quite a long type with an object, that is more an indication of that object being interesting than the busy type becoming more greedy in their visit. Table 3 below shows the total amount of time, in seconds, that the fifty tracked visitors have spent at the twelve targeted objects at MIT's Robots and Beyond exhibit. A simple histogram plot

[Fig. 4] shows which of these objects are interesting/neutral/boring, always in a probabilistic sense. The tallest bins are from objects 5,9,11, which are therefore the most interesting. The shortest bins are from objects 1,4,12 that are therefore boring. All remaining objects shall be considered neutral.

Table 3. Total time that the tracked visitors have spent at the 12 selected objects

Intro 1	Lisp 2	Minsky Arm 3	Robo Arm 4	Falcon 5	Phantom 6	Cogs Head 7	Quad 8	Uniroo 9	Dext Arm 10	Kismet 11	Baby Doll 12
588	755	719	627	986	748	771	694	859	700	1025	548

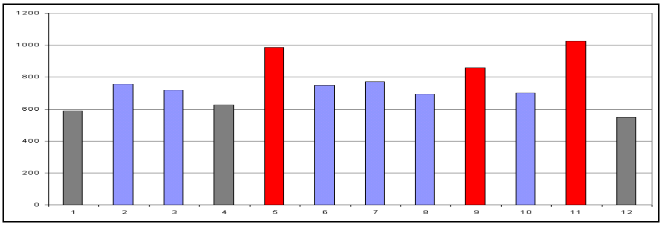


Fig. 4. Histogram of overall time, in seconds, that visitors spend at the 12 selected objects

Based on the information above, a set of priors of the twelve objects is given by Table 4: The conditional probabilities p(location | object, visitor), initially assigned by the “expert” (system modeler, museum curator), and later refined in the learning phase, described in the next section, are shown in Table 5. The initial object, and transition probabilities from one object to the next, are given by Table 6. The meaning of the transition probability table is quite important for the system. The values along the diagonal describe how much past history matters in determining if a visitor belongs to a type. For example if the values along the diagonal are high (0.9) and the visitor switches behavior from let say greedy to selective, the system will have some *inertia* in identifying the new type. It will take the system a few time steps to catch up with the new visitor behavior. This makes sense if there are many objects in the exhibit. If instead there are only a few objects in the exhibit, such as in the MIT’s Robots and Beyond targeted exhibit, it is desirable to have a system which adapts fast to the visitor’s behavior and does not tend to “stick to the first impression” as before. In this case it is preferable to have lower values along the diagonal (less than 0.6). The transition probabilities are in this model the same for all twelve objects. However these could be also learned if more training data was available. For example visitors could have a tendency to skip certain objects, or visit them in a different order than what the exhibit designer has laid out. Alternatively, the transition probabilities would highlight groups of object and show that for example the “Sensors” section of the exhibit turns out to be more interesting than the “Movement” section. This information would be reflected in the transition tables. From the visitor tracking data gathered at the MIT Museum we observed people visiting objects one after the next in a linear sequence. This however does not limit the validity of the procedure here described. If a visitor returns to an object the system will select content based on the value of the visitor node at that time in the history of the visit. For example, during the second visit to an object the visitor could be “greedier” than when he/she first stopped in that location, because of his/her behavior at the subsequently visited objects. Content will then be selected according to the “new” current value of the visitor node, and could be the same or different than in the previous stop at that same object.

Table 4. Priors for the 12 selected objects derived from the visitor tracking data

	1	2	3	4	5	6	7	8	9	10	11	12
Neutral	0.23	0.34	0.28	0.25	0.18	0.32	0.36	0.24	0.27	0.26	0.16	0.2
Interesting	0.23	0.33	0.36	0.25	0.64	0.34	0.32	0.38	0.46	0.37	0.68	0.2
Boring	0.54	0.33	0.36	0.5	0.18	0.34	0.32	0.38	0.27	0.37	0.16	0.6

Table 5. Conditional Probability Tables for the Location Nodes

	neutral			Interesting			boring		
	busy	greedy	selective	busy	greedy	selective	busy	greedy	selective
skip	0.2	0.1	0.4	0.1	0.05	0.1	0.35	0.2	0.65
short	0.7	0.1	0.2	0.6	0.05	0.3	0.6	0.2	0.15
long	0.1	0.8	0.4	0.3	0.9	0.6	0.05	0.6	0.2

Table 6. Priors for the 12 selected objects and transition probabilities

P(O1)		P(Oj | Oi)	neutral	interesting	boring
neutral	0.333	neutral	0.6	0.2	0.2
interesting	0.333	interesting	0.2	0.6	0.2
boring	0.333	boring	0.2	0.2	0.6

To test the model, I introduced evidence on the duration nodes, thereby simulating its functioning during the museum visit [fig 5]. I have included results below, limited to two time slices, for the limited space available on paper. The reader can verify that the system gives plausible estimates of the visitor type, based on the evidence introduced in the system. The posterior probabilities in this and the subsequent models are calculated using Hugin, (www.hugin.com) which implements the Distribute Evidence and Collect Evidence message passing algorithms on the junction tree.

Test case 1. The visitor spends a short time both with the first and second object —> the network gives the highest probability to the busy type (0.8592)

Test case 2. The visitor spends a long time both with the first and second object —> the network gives the highest probability to the greedy type (0.7409)

Test case 3. The visitor spends a long time with the first object and skips the second object —> the network gives the highest probability to the selective type (0.5470)

5 Data Preprocessing and Labeling

The first task before model training, is to assign labels to the tracking data gathered at the museum [Table 2]. For example, for the targeted exhibit, I need to decide whether a stop of 18 seconds should be considered 'short' or 'long'. Various classification techniques can be used: the classification problem addressed is simple enough for unsupervised classification techniques, such as K-means, to be effective. To have reliable results, I used two different classification techniques: one is the popular K-means classification procedure [11]. The other consists in simply plotting histograms of the data, and finding a threshold between 'short' and 'long' stops by histogram intersection. The classes of data that are needed are only two: 'short' and 'long', as 'skip' is easily identifiable as a stop of zero seconds duration. I used both methods and compared results, as discussed below.

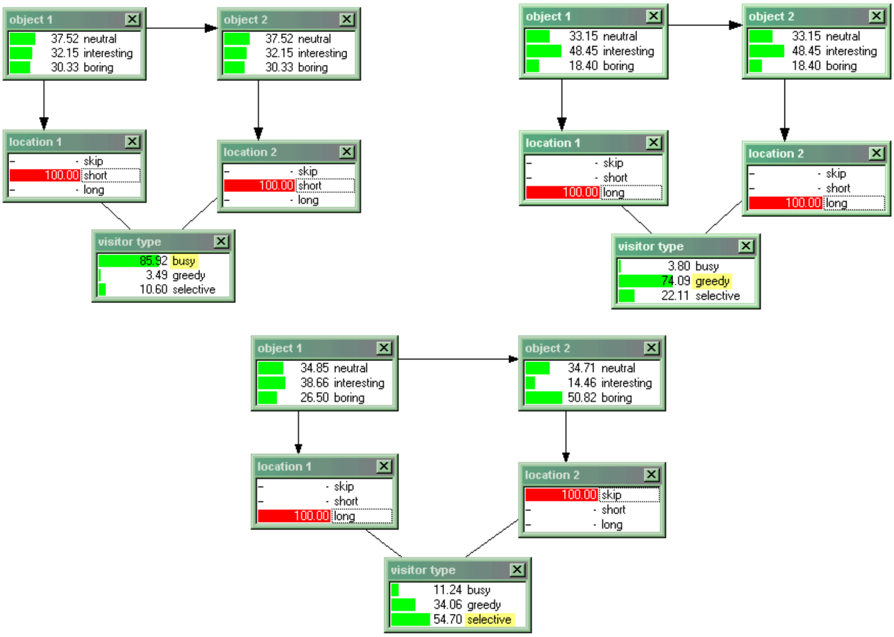


Fig. 5. Testing the model: posterior probabilities after evidence is introduced and inference is performed: for the busy and greedy cases

An important step before data clustering is performing data quantization, when necessary. The data often carries more precision than needed, and may therefore include too many categories with too much precision. This can be a problem as extra subdivisions can hide trends. Preprocessing reduces the number of variables without jeopardizing the results.. I quantized the data in groups of five, and obtained the histogram shown in Fig. 6, which clearly exhibits an exponential decay data trend. This means roughly that visitors globally tend to make many more short than long stops at this exhibit.

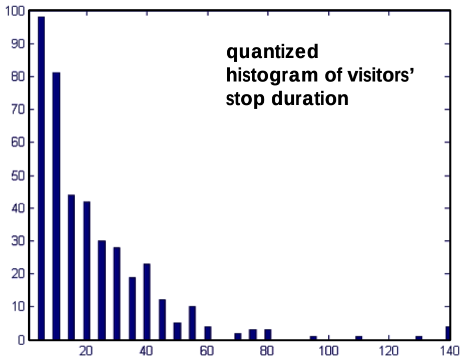


Fig. 6. Quantized histogram for the duration of visitors' stops

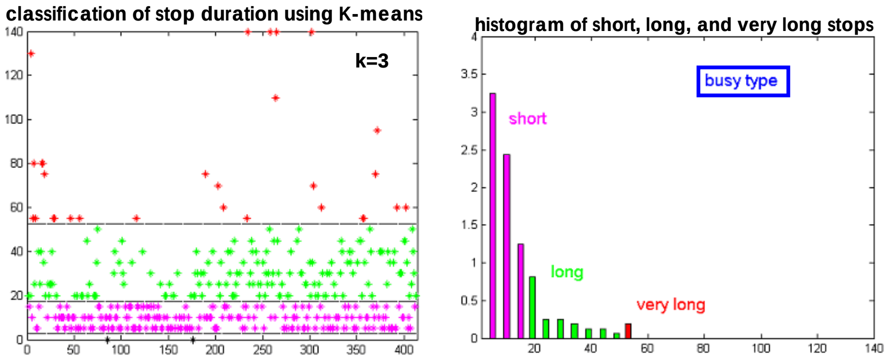


Fig. 7. K-means classification of stop durations for the greedy, busy and selective types and histogram of stop durations for the busy type

I grouped the observed visitors according to their type: I observed a total of 3 greedy types, 16 busy types, and 31 selective types. These types contribute differently to the quantized histogram. I performed k-means analysis of the visitor tracking data, with k=3, to allow the extra third cluster to collect all the samples which do not fall into the 'short' or 'long' category. I also set the center cluster at start to the approximate initial values of: 10, 30, and 60 seconds respectively. K-means gives three clusters centered respectively in 9, 30, and 79, and with ranges 1-19, 20-54, and 55-140. I labeled these clusters as 'short', 'long' and 'very long'. K-means also gives a threshold of 19 seconds as separator between short and long stops, for visitors at MIT Museum's Robots and Beyond exhibit [Fig. 7]. With the knowledge of these ranges, I have plotted separate normalized histograms of stop durations for the three types [Fig. 7], by color coding the bins corresponding to the different stop-duration ranges found. These plots confirm our intuition that a greedy type tends to make several long stops (and skips only very few object); a busy type makes many short stops; and the selective type makes several long stops (and skips many objects). For further discussion I will merge the long and very long clusters into one 'long' cluster, ranging from 20 to 140 seconds.

6 Learning from the Data: Model Validation

I grouped the visitor tracking data gathered at the museum in two groups: a group from which I trained the proposed Bayesian network (i.e. I learned the parameters of the network from the data) and a control group on which I performed out-of-sample testing to validate the predictive power of the model. For robustness I split the data in a training group and a control group in three different ways, repeated the learning procedure three times, and compared the obtained results. For simplicity, in the training phase of this research, I assigned only two trivial states to the object node: Visited and Not Visited. Given that the system is learning the conditional probability table of the location nodes, conditioned on the visitor and object nodes, i.e. $p(L|O,V)$, in the case with object nodes with three states (Neutral, Interesting, Uninteresting) we would have to learn 27 parameters (3 location states x 3 object states x 3 visitor states) instead of 18 (3x2x3) to train the network with the data. To learn the 9 $p(L|O,V)$ parameters (only 9 to learn for the visited nodes) I applied the Expectation Maximization algorithm [12].

I used the new learned parameters to test how well the Bayesian network performs in identifying the other half of the tracked user types. In each case I took the original set of 50 visitor tracking data, and split it randomly in two parts, each composed of 25 subjects. I used the first half of the subjects as training data, and the remaining half as test data. For robustness I performed this operation three times, each time with a different random subdivision of the original data set in half. I then, for each of the three learned conditional probability tables for the location nodes, substituted the original conditional probability table with the new learned parameters. Then, for each of the 25 visitor data (row) in the test group, I introduced the stop duration for the 12 tracked objects as evidence in the network, and calculated the posterior probability on the visitor nodes. I compared the visitor's busy/greedy/selective state with the highest probability, with the label assigned to the visitor behavior in the test file. When the original labeled data coincided with the posterior with the highest probability I considered this a 'success', otherwise a 'miss'. For the three test cases, each made of 25 cases, I obtained respectively 25, 24, and 25 successes. There was only one miss, with the following test data (short, skip, short, short, short, skip, short, skip, skip, long, long, skip) which the network classified as 'busy' while it was labeled 'selective'. The misclassified visitor had therefore a behavior at the dividing boundary between busy and selective, as they have very close absolute errors. I would attribute the error to the human who observed the visitors' behavior at the museum, and labeled that particular's visitor behavior wrongly because of the ambiguity of that test case. Given the high success rate of this learning/test procedure, which can be quantified as 74/75= 0.987, for further visitor classification with the given Bayesian network, I have performed EM learning on all 50 visitors obtaining the final learned parameters in Table 7. The values in this table can be seen as a data-grounded refinement of the expert defined conditional probability table [Table 1].

Table 7. New learned probability table, for all 50 visitor data

Final Conditional Probability Table p(L|O,V)			
	skip	**short**	**long**
Busy	0.25	0.625	0.125
Greedy	0.14	0.22	0.64
Selective	0.36	0.3	0.34

7 Editing Stories for Different Visitor Types and Profiles

Sto(ry)chastics works in two steps. The first is user type estimation as previously described. The next step is to assemble a mini-story for the visitor, relative to the object he/she is next to. Most of the audio-visual material available for art and science documentaries tends to fall under a set of characterizing topics. After an overview of the audio-visual material available at MIT's Robots and Beyond exhibit, the following content labels, or bins were identified to classify the component video clips:

1. Description of the artwork: what it is, when it was created (answers: *when, where, what*) 2. Biography of author: anecdotes, important people in artist's life (answers: *who*) 3. History of the artwork: previous relevant work of the artist 4. Context: historical, what is happening in the world at the time of creation 5. Process: particular techniques used or invented to create the artwork (answers: *how*) 6. Principle: philosophy or school of thought the author believes in when creating the artwork (answers: *why*) 7.Form and Function: relevant style, form and function which contribute to explain the artwork. 8.Relationships: how is the artwork related to other artwork on display 9. Impact: the critics' and the public's reaction to the artwork.

A similar approach to documentary as a composition of segments belonging to different themes, has been developed by Houbart in her work which edits a documentary based on the viewer's theme preferences, as an off-line process [13]. The difference between Houbart's work and what the museum wearable does is that the museum wearable performs editing in real time, using sensor input and Bayesian network modeling to figure out the user's preferences (type). This project required a great amount of editing to be done by hand (non automatically) in order to segment the two hours of video material available for the Robots and Beyond Exhibit at the MIT museum in the smallest possible complete segments. After this phase, all the component video clips were given a name, their length in seconds was recorded into the system, and they were also classified according to the list of bins described above. The classification was done probabilistically, that is each clip has been assigned a probability (a value between zero and one) of belonging to a story category. The sum of such probabilities for each clip needs to be one. The result of the clip classification procedure, for a subset of available clips, is shown in Table 8.

To perform content selection, conditioned on the knowledge of the visitor type, the system needs to be given a list of available clips, and the criteria for selection. There are two competing criteria: one is given by the total length of the edited story for each object, and the other is given by the ordering of the selected clips. The order of story

Table 8. Examples of segments cut from the video documentation available for the MIT Museum's Robots and Beyond Exhibit. All segments have been assigned a set of probabilities which express their relevance with respect to nine relevant story themes or categories

CATEGORIES/ TITLES LENGTH IN SECS	**Kismet Cynt.bio 067**	**Kismet social 183**	**Kismet intro 066**	**Kismet develop 066**	**Cog drum 083**	**Cog future 043**	**Cog intro 041**	**Cog history 051**
Description	0	0.1	0.3	0.1	0.3	0	0.8	0
History	0	0	0	0.2	0	0	0	0.5
Context	0	0.1	0	0	0	0	0	0
Biography	0.9	0	0	0	0	0	0	0.1
Process	0	0.3	0.3	0.1	0.6	0	0.2	0
Principle	0	0.5	0.3	0.2	0.1	1	0	0.2
Form & Function	0	0	0.1	0.4	0	0	0	0.2
Relationships	0.1	0	0	0	0	0	0	0
Impact	0	0	0	0	0	0	0	0
Total P	1	1	1	1	1	1	1	1

segments guarantees that the curator's message is correctly passed on to the visitor, and that the story is a "good story", in that it respects basic cause-effect relationships and makes sense to humans. Therefore the Bayesian network described in the earlier needs to be extended with additional nodes for content selection [Fig. 8]. The additional "good story" node, encodes, as prior probabilities, the curator's preferences about how the story for each object should be told. To reflect these observations the Bayesian network is extended to be an influence diagram [14]: it will include decision nodes, and utility nodes which guide decisions. The decision node contains a list of all available content (movie clips) for each object. The utility nodes encode the two selection criteria: length and order. The utility node which describes length, contains the actual length in seconds for each clip. The length is transcribed in the network as a positive number, when conditioned on a preference for long clips (greedy and selective types). It is instead a negative length if conditioned on a preference for short content segments (busy type). This is because a utility node will always try to maximize the utility, and therefore length is penalizing in the case of a preference for short content segments. The utility node which describes order, contains the profiling of each clip into the story bins described earlier, and listed in Table 8, times a multiplication constant used to establish a balance of power between "length" and "order". Basically order here means a ranking of clips based on how closely they match the curator's preferences expressed in the "good story" node. By means of probability update, the Bayesian network comes up with a "compromise" between length and order and provides a final ranking of the available content segments in the order in which they should be played [Fig. 9].

As an example, Table 9 shows two possible definition of "good story", by two different curators, called for easy reference, Frank and Liz. What the numbers in the table say is that Frank believes that a good museum story should start with an extensive object description, followed by biographical information about its creator. Next, explanation about the process of creation should be given, accompanied by a history of previous versions or sketches of the same object, and elements of form and function. Of less importance are the relationship of the object to other objects on display, the guiding philosophical principles which have led to its creation, its impact of the public and the art critics, and what was happening in the world at the time of creation. Liz thinks differently than Frank. She believes that a good museum story should be based on the creator's profile and biographical information, and that these elements should have the priority. Explaining to the public what previous artwork has led to the creation of the object they are looking at is also important. Information about Form and Function, accompanied by a more detailed description of the object should follow. All the other themes, or story bins, are of secondary importance. The different results of these two rather opposite views of story are:

Table 9. Two different views of what is a "good story" according to curators Frank and Liz

Story bins or CATEGORIES		curator Frank	Curator Liz
Description	DSC	0.3	0.1
History	HST	0.1	0.19
Context	CTX	0.03	0.03
Biography	BIO	0.16	0.21
Process	PRC	0.14	0.02
Principle	PNC	0.06	0.2
Form & Function	FAF	0.09	0.12
Relationships	REL	0.08	0.05
Impact	IMP	0.04	0.08
Total P		1	1

These possible choices of Liz and Frank are given only to provide examples how the system works. The same curator could actually choose a different combination of weights for the good story node for a different exhibit, as the message carried by an exhibits changes with its content. A third curator, let say Nancy, could argue that, in her view, a story should always start with the shortest possible introductory and descriptive clip, followed by a longer description clip, followed by a segment which describes the creative process and so on. Her preferences can be easily accommodated by having the system always select the shortest description clip in first place, and then using the segment ranking provided by the Bayesian network for the following segments. The Bayesian network leaves therefore plenty of choice to the exhibit designer, curator, and storyteller in terms of easy access to the knobs of the virtual storytelling machine without the need to calculate in advance all possible combinations given by all the knob values.

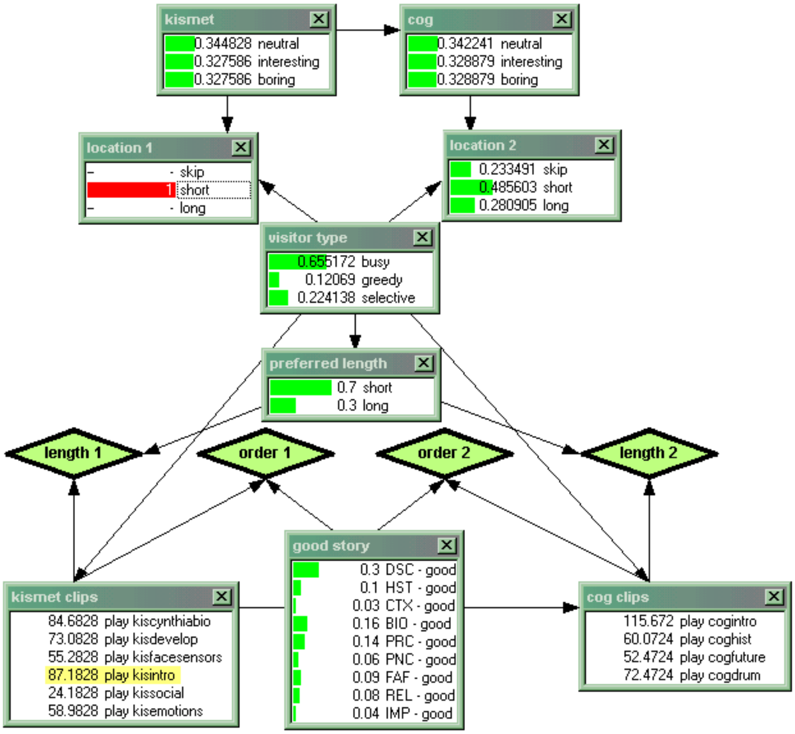


Fig. 8. Bayesian network for visitor type estimation extended to perform content selection

Fig. 9. Storyboards from various video clips shown on the museum wearable's display at MIT Museum's Robots and Beyond Exhibit

8 Summary of Accomplishments and Future Directions

The main contribution of this research is to show that (dynamic) Bayesian networks are a powerful modeling technique to couple inputs to outputs for real time sensor-driven multimedia audiovisual stories, such those that are triggered by the body in motion in a sensor-instrumented interactive narrative space. Sto(ry)chastics has implications both for the human author (designer/curator) which is given a flexible modeling tool to organize, select, and deliver the story material, as well as the audience, which receives personalized content only when and where it is appropriate. Sto(ry)chastics proposes an alternative to complex centralized interactive entertainment programs which simply read sensor inputs and map them to actions on the screen. These systems rigidly define the interaction modality with the public, as a consequence of their internal architecture. Sto(ry)chastics performs probabilistic reasoning under uncertainty in real time to identify the visitor's type. It then delivers an audiovisual narration to the visitor as a function of the estimated type, interactively in time and space.

The model has been tested and validated on observed visitor tracking data using the EM algorithm. The interpretation of sensor data is robust in the sense that it is probabilistically weighted by the history of interaction of the participant as well as the nodes which represent context. Therefore noisy sensor data, triggered for example by external or unpredictable sources, is not likely to cause the system to produce a response which does not "make sense" to the user.

An experimentation phase at the museum should follow the current research. A procedure would have to be set to establish if and how the museum wearable does actually enhance learning and entertainment at an exhibit, or how the content shown does actually match the visitor's preferences. Also, more visitor tracking data would need to be gathered at the museum site, to eventually infer more visitor types than the ones described in this document, and compare them with the more sophisticated visitor typologies discussed in the museum literature.

Acknowledgements

I would like to especially acknowledge and thank Yuan Qi for his insightful suggestions and help on this research. Tom Minka provided advice on the model validation section of this paper and originally introduced me to Bayesian Networks. Jim Rehg and Bill Freeman encouraged me in the initial steps, and provided references and guidance. I am also grateful to Neil Gerschenfeld who has been the strongest supporter of this work. Walter Bender provided logistics and financial support. Kent Larson, Ron McNeil, Glorianna Davenport offered various suggestions and support. Work on wearable computing was originally developed with Sandy Pentland. Several undergraduate collaborators have contributed to the Museum Wearable: Eric Hilton, Chin Yan Wong, Anjali, D'Oza, Sarah Mendelowitz, Manuel Martinez, Tracie Lee, Jeff Bender. Several calculations were performed with HUGIN's real time C devel-

opment environment (http://www.hugin.com) and Kevin Murphy's MATLAB based BNT (Bayesian Network Toolkit)
(http://www.cs.berkeley.edu/~murphyk/Bayes/bnt.html).

References

1. Sparacino F., Davenport G., Pentland A. "Media in Performance." IBM Systems Journal, Vol. 39, Nos. 3 & 4, 2000, pp 479-510.
2. Schiele, B., Oliver, N., Jebara, T., and Pentland, A. "An Interactive Computer Vision System, DyPERS: Dynamic and Personal Enhanced Reality System." International Conference on Computer Vision Systems, Gran Canaria, Spain, Springer, pp. 51-65, 1999
3. Feiner, S., MacIntyre, B., Hollerer, T., and Webster, A. "A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment." In: Proceedings of the International Symposium on Wearable Computers, pp. 74-83. Los Alamitos, CA, USA: IEEE Computer Society, 1997.
4. Pearl, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, CA, 1988.
5. Jordan M.I., editor. Learning in Graphical Models. The MIT Press, 1999.
6. Pynadath D. V. and Wellman M. P. "Accounting for context in plan recognition, with application to traffic monitoring." Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, 1995, pp. 472-481, 1995.
7. Albrecht, D.W., Zukerman, I., Nicholson, A.E., and Bud, A. "Towards a bayesian model for keyhole plan recognition in large domains." In Jameson, A.; Paris, C.; and Tasso, C., eds., Proceedings of the Sixth International Conference on User Modeling (UM '97), pp. 365-376. Springer, 1997
8. Maes, P. Intelligent Software. Intelligent User Interfaces.pp41-43 1997.
9. Koller, D. and Pfeffer, A. "Probabilistic Frame-Based Systems." In: Proceedings of the Fifteenth National Conference on Artificial Intelligence; Madison, Wisconsin; July 1998.
10. Sparacino, F. The Museum Wearable: real-time sensor-driven understanding of visitors' interests for personalized visually-augmented museum experiences. In: Proceedings of Museums and the Web (MW2002), April 17-20, Boston, 2002
11. Therrien, C.W. Decision, Estimation, and Classification. John Wiley and Sons, 1989.
12. Dempster, A.P., Laird, N.M., and Rubin, D.B. "Maximul Likelihood from incomplete data via the EM algorithm." Journal of the Royal Statistical Society, B 39 (1), 1-38, 1977.
13. Houbart, G. Viewpoints on Demand: Tailoring the Presentation of Opinions in Video. MIT Masters Thesis, 1994
14. Howard, R.A., and Matheson, J. E. "Influence Diagrams." In: Applications of Decision Analysis, volume 2, eds. R.A. Howard and J.E. Matheson, 721-762, 1981.

Inferring High-Level Behavior from Low-Level Sensors

Donald J. Patterson, Lin Liao, Dieter Fox, and Henry Kautz

University Of Washington
Department of Computer Science and Engineering
Seattle, WA, USA
{djp3,liaolin,fox,kautz}@cs.washington.edu

Abstract. We present a method of learning a Bayesian model of a traveler moving through an urban environment. This technique is novel in that it simultaneously learns a unified model of the traveler's current mode of transportation as well as his most likely route, in an unsupervised manner. The model is implemented using particle filters and learned using Expectation-Maximization. The training data is drawn from a GPS sensor stream that was collected by the authors over a period of three months. We demonstrate that by adding more external knowledge about bus routes and bus stops, accuracy is improved.

1 Introduction

A central theme in ubiquitous computing is building rich predictive models of human behavior from low-level sensor data. One strand of such work concerns tracking and predicting a person's movements in outdoor settings using GPS [1,2,3,4]. But location is only one small part of a person's state. Ideally we would want to recognize and predict the high-level intentions and complex behaviors that cause particular physical movements through space. Such higher-order models would both enable the creation of new computing services that autonomously respond to a person's unspoken needs, and support much more accurate predictions about future behavior at all levels of abstraction.

This paper presents an approach to learning how a person uses different kinds of transportation in the community. We use GPS data to infer and predict a user's transportation *mode*, such as walking, driving, or taking a bus. The learned model can predict mode transitions, such as boarding a bus at one location and disembarking at another. We show that the use of such a higher-level transportation model can also increase the accuracy of location prediction, which is important in order to handle GPS signal loss or preparing for future delivery of services.

A key to inferring high-level behavior is fusing a user's historic sensor data with general commonsense knowledge of real-world constraints. Real-world constraints include, for example, that buses only take passengers on or off at bus stops, that cars are left in parking lots, and that cars and buses can only travel on streets, *etc.*. We present a unified probabilistic framework that accounts for both sensor error (in the case of GPS, loss of signal, triangulation error, or multi-path propagation error) and commonsense rules.

Although this work has broad applications to ubiquitous computing systems, our motivating application is one we call the Activity Compass, a device which helps guide a cognitively impaired person safely through the community [5]. The system notes when

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 73–89, 2003.

the user departs from a familiar routine (for example, gets on the wrong bus) and provides proactive alerts or calls for assistance. The Activity Compass is part of a larger project on building cognitive assistants that use probabilistic models of human behavior [6].

Our approach is built on recent successes in particle filters, a variant of Bayes filters for estimating the state of a dynamic system [7]. In particular we show how the notion of graph-constrained particle filtering introduced in [8] can be used to integrate information from street maps. Extensions to this technique include richer user transportation state models and multiple kinds of commonsense background knowledge. We introduce a three-part model in which a low-level filter continuously corrects systematic sensor error, a particle filter uses a switching state-space model for different transportation modes (and further for different velocity bands within a transportation mode), and a street map guides the particles through the high-level transition model of the graph structure. We additionally show how to apply Expectation-Maximization (EM) to learn typical motion patterns of humans in a completely unsupervised manner. The transition probabilities learned from real data significantly increase the model's predictive quality and robustness to loss of GPS signal.

This paper is organized as follows. In the next section, we summarize the derivation of graph-based tracking starting from the general Bayes filter, and show how it can be extended to handle transportation mode tracking. Then, in Sect. 3, we show how to learn the parameters of the tracking model using EM. Before concluding in Sect. 5, we present experimental results that show we can learn effective predictive models of transportation use behavior.

2 Tracking on a Graph

Our approach tracks a person's location and mode of transportation using street maps such as the ones being used for route planning and GPS-based car tracking. More specifically, our model of the world is a graph $G = (V, E)$ which has a set V of vertices and a set E of directed edges. Edges correspond to straight sections of roads and foot paths, and vertices are placed in the graph to represent either an intersection, or to accurately model a curved road as a set of short straight edges. To estimate the location and transportation mode of a person we apply Bayes filters, a probabilistic approach for estimating the state of a dynamic system from noisy sensor data. We will now briefly describe Bayes filters in the general case, show how to project the different quantities of the Bayes filter onto the structure represented in a graph, and then discuss our extensions to the state space model.

2.1 Bayesian Filtering on a Graph

Bayes filters address the problem of estimating the state x_t of a dynamical system from sensor measurements. Uncertainty is handled by representing all quantities involved in the estimation process using random variables. The key idea of Bayes filters is to recursively estimate the posterior probability density over the state space conditioned on the data collected so far. The data consists of a sequence of observations $z_{1:t}$ and the posterior over the state x_t at time t is computed from the previous state x_{t-1} using the following update rule (see [7,9] for details):

$$p(x_t \,|\, z_{1:t}) \propto p(z_t \,|\, x_t) \int p(x_t \,|\, x_{t-1})\; p(x_{t-1} \,|\, z_{1:t-1}) dx_{t-1} \qquad (1)$$

The term $p(x_t \mid x_{t-1})$ is a probabilistic model of the object dynamics, and $p(z_t \mid x_t)$ describes the likelihood of making observation z_t given the location x_t.

In the context of location estimation, the state, x_t, typically describes the position and velocity of the object in 2D-space. When applying Bayesian filtering to a graph, the state of an object becomes a triple $x_t = \langle e, d, v \rangle$, where $e \in E$ denotes on which edge the object resides, d indicates the distance of the object from the start vertex of edge e, and v indicates the velocity along the edge [8]. The motion model $p(x_t \,|\, x_{t-1})$ considers that the objects are constrained to motion on the graph and may either travel along an edge, or, at the endpoint of the edge, switch to a neighboring edge. To compute the probability of motion from one edge to another, the graph is annotated with transition probabilities $p(e_j \mid e_i)$, which describe the probability that the object transits to edge e_j given that the previous edge was e_i and an edge transition took place. Without other knowledge, this probability is a uniform distribution over all neighboring edges of e_i.

Our work builds on graph-based Bayesian tracking by hierarchically extending the state model. We add a higher level of abstraction which contains the transportation information and a lower level sensor error variable. The resulting state x_t consists of the variables shown in Fig. 1. The presence of a bus stop near the person is given by the binary variable b_t, and the presence of a parking lot is modeled by c_t. The mode of transportation, denoted m_t, can take on one of three different values

$$m_t \in \{BUS, FOOT, CAR\}.$$

v_t denotes the motion velocity, and the location of the person at time t is represented by $l_t = \langle e, d \rangle$. o_t denotes the expected sensor error, which in our current model compensates for systematic GPS offsets. Finally, at the lowest level of the model, raw GPS sensor measurements are represented by gps_t.

Tracking such a combined state space can be computationally demanding. Fortunately, Bayes filters can make use of the independences between the different parts of the tracking problem. Such independences are typically displayed in a graphical model like Fig. 1. A dynamic Bayes net [10,11], such as this one, consists of a set of variables for each time point t, where an arc from one variable to another indicates a causal influence. Although all of the links are equivalent in their causality, Fig. 1 represents causality through time with dashed arrows. In an abstract sense the network can be as large as the maximum value of t (perhaps infinite), but under the assumption that the dependencies between variables do not change over time, and that the state space conforms to the first-order Markov independence assumption, it is only necessary to represent and reason about two time slices at a time. In the figure the slices are numbered $t-1$ and t. The variables labeled gps are directly observable, and represent the position and velocity readings from the GPS sensor (where a possible value for the reading includes "loss of signal"). All of the other variables — sensor error, velocity, user location, mode, and the presence of a parking lot or bus stop location — are hidden variables whose values must be inferred from the raw GPS readings.

The dependencies between the nodes in Fig. 1 can be quite complex. The GPS reading at each time point is influenced by the local sensor error and the user's actual velocity

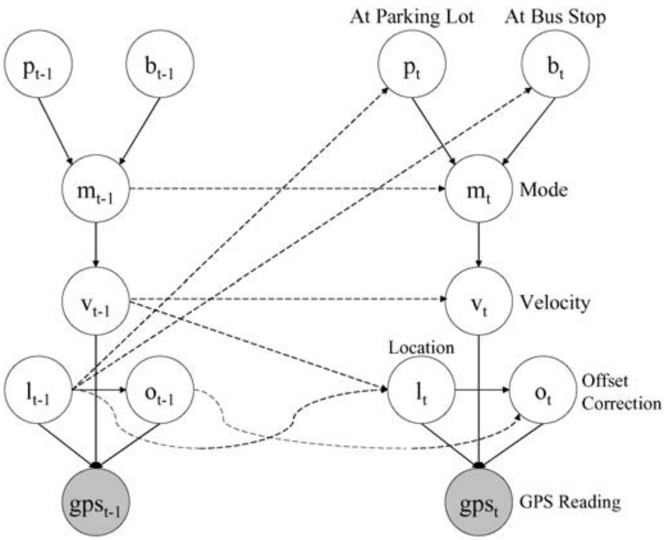


Fig. 1. Two-slice dynamic Bayes net model of the transportation domain, showing dependencies between the observed and hidden variables. Observed variables are shaded. Intra-temporal causal links are solid, inter-temporal links are dashed.

and location. The location at time t only depends on the person's previous location and the motion velocity. Note that GPS is explicitly not considered to provide the true user location; urban interference, map reference point errors, GPS error and sensor failure all cause the true location to be a hidden variable. The sensor offset correction node o_t is used to reason about errors in the GPS readings which are systematic over time and location. This node maintains a probability distribution over corrections to the GPS signal that are caused by multi-path propagation error and/or dynamic satellite geometry. The node updates its belief state by comparing GPS readings to the street map to gradually adjust to local variations in signal offset.

A more complex relationship governs how the mode of transportation influences the instantaneous velocity. The influence of mode on velocity is complicated by the fact that the range of possible instantaneous velocities for each mode overlap. For example, movement at 7 km/hr may be a brisk walk or a slowly moving car or bus. To simplify the relationship between mode and velocity we model the continuous velocities using the Gaussian mixture shown in Fig. 2. A separate unsupervised Expectation-Maximization (EM) process determined the parameters of these probability densities using real velocity data. Our model assumes that velocities are drawn randomly from these Gaussians, where the probability of drawing from a particular Gaussian depends on the mode. For example, the walking mode draws a speed from the left-most cluster with probability one. In the bus mode, the person has a $1/3$ chance of being in each of the three slowest velocity clusters. In our current approach, the probabilities for the Gaussians in the different transportation modes were set manually based on external knowledge. Learning the weights of the mixture components depending on the transportation mode (and eventually location) is left for future research.

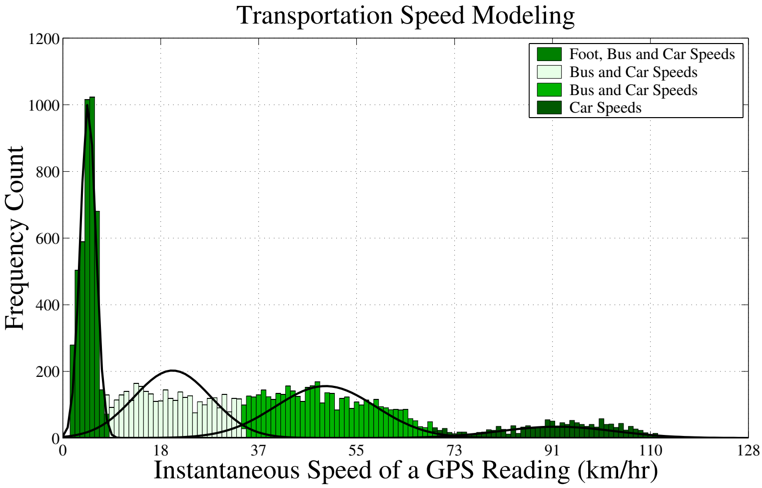


Fig. 2. Gaussian mixture model for the dependency of transportation mode on velocities. The Gaussians were learned using EM based on previously collected velocity data. The frequencies of the raw velocity values are indicated by the bins. Different transportation modes are modeled by sampling with different probability from the four Gaussians.

In our model, the motion mode at time t only depends on the previous mode and the presence of a parking lot or bus stop. For example, the person can only get on a bus if the node b_t indicates the presence of a bus stop. The values of the bus stop and parking lot nodes depend on the location of the person, as indicated by the arrows in the model shown in Fig. 1. Learning mode and location transition probabilities is an important aspect of our approach and will be discussed in Sect. 3.

2.2 Particle Filter Based Implementation

Particle filters provide a sample-based implementation of general Bayes filters [7]. They represent posterior distributions over the state space with temporal sets, S_t, of n weighted samples:

$$S_t = \{\langle x_t^{(i)}, w_t^{(i)} \rangle \mid i = 1, \ldots, n\}$$

Here each $x_t^{(i)}$ is a sample (or state), and the $w_t^{(i)}$ are non-negative numerical factors called *importance weights*, which sum up to one. Like Kalman filters, particle filters apply the recursive Bayes filter update to estimate posteriors over the state space, but unlike Kalman filters, particle filters are not restricted to unimodal posterior distributions[1]. The basic particle filter updates the posterior according to the following sampling procedure, often referred to as sequential importance sampling with re-sampling (SISR, see also [7]):

[1] We consider multi-hypothesis tracking to be a viable alternative to our particle filter based implementation. Multi-hypothesis tracking overcomes the restrictive assumption of the plain Kalman filter by estimating a state using multiple Kalman filters [12]. An implementation of such an approach will be part of our future research.

- **Sampling:** Draw n samples $x_{t-1}^{(i)}$ from the previous set and generate n new samples $x_t^{(j)}$ using the distribution

$$p(x_t \mid x_{t-1}).$$

The new samples now represent the density given by the product

$$p(x_t \mid x_{t-1})p(x_{t-1} \mid z_{1:t-1})$$

This density is the so-called *proposal distribution* used in the next step.

- **Importance sampling:** Assign each sample $x_t^{(j)}$ an importance weight according to the likelihood of the observation, z_t, given the sample,

$$w_t^{(j)} = p(z_t \mid x_t^{(j)}).$$

- **Re-sampling:** Multiply / discard samples by drawing samples with replacement according to the distribution defined through the importance weights $w_t^{(j)}$.

It can be shown that this procedure in fact approximates the Bayes filter update (1), using a sample-based representation [7,13].

The application of particle filters to the problem of location and mode estimation using the network shown in Fig. 1 is rather straightforward. Each particle $x_t^{(i)}$ represents an instantiation of the random variables describing the transportation mode m_t, the location l_t, and the velocity v_t. The parking lot and bus stop variables p_t and b_t are extracted from each sample location l_t. Finally, o_t is determined globally for all particles by estimating the offset between GPS readings and the street map. The update steps of the particle filter can be implemented as follows. The temporal sampling step corresponds to advancing each particle according to the motion model: First the transportation mode is chosen according to the previous transportation mode and the presence of bus stops or parking lots. This gives us m_t. Then we randomly pick a velocity from the velocity model for the specific mode m_t. The velocity is used to advance the position of the person on the graph. If the sampled velocity implies a transition to another edge, the next edge e_t is drawn with probability $p(e_t \mid e_{t-1}, m_t)$ (see [8] for more information on edge transitions). After these sampling steps, the resulting states represent the predicted location, velocity, and transportation mode. The importance sampling step is implemented by weighting each sample according to the likelihood of observing the current signal from the GPS sensor given the new location of the sample. The re-sampling step of the particle filter algorithm does not have to be changed.

3 Parameter Learning

One of the advantages of modeling the world with a graph is the ability to record behavioral data about edge transitions. The discrete nature of such transitions facilitates unsupervised learning of hierarchical model parameters. We have an intuitive prior expectation of how state transitions occur between and within edges: edge transitions occur uniformly among the edge's neighbors, and mode transitions vary according to the presence of a bus stop or parking lot.

Learning in this context means adjusting the model parameters to better fit the training data, typically to better model an individual user or the environment. Learning parameters for specific individuals captures idiosyncratic motion patterns — the movements the user commonly makes, as opposed to the logically possible set of movements. Since our model also includes transportation mode, learning also means changing our prior expectations about which edges mode transitions occur on. Bus stops and parking locations are conceptual locations where mode transitions may occur. Our model enables learning of the commonly used subset of these locations, to highlight where a user frequently parks her car, for example. The learned model supports better tracking and prediction than the prior model, and is the foundation upon which high-level understanding of the user's behavior is built.

We now describe how to learn the parameters of our graph model using data collected by a person moving through the community. Our motivating application of the Activity Compass forces us to learn the transportation modes in an *unsupervised manner*. When deployed, Activity Compass users must not be required, for example, to keep a diary for several weeks of their transportation modes in order to create a supervised training set. Hence, the most obvious difficulty is that we have to learn the motion model based solely on a map and a stream of non-continuous and noisy GPS sensor data.

A general approach for solving such learning problems is the well-known Expectation-Maximization (EM) algorithm [14,15]. In our application, EM is based on the observation that learning the model parameters would be easy *if* we knew the person's true location and transportation mode at any point in time. Unfortunately, location and transportation mode are hidden variables, *i.e.* they cannot be observed directly but have to be inferred from the raw GPS measurements. EM solves this problem by iterating between an Expectation step (E-step) and a Maximization step (M-step). In a nutshell, each E-step estimates expectations (distributions) over the hidden variables using the GPS observations along with the current estimate of the model parameters. Then in the M-step the model parameters are updated using the expectations of the hidden variables obtained in the E-step. The updated model is then used in the next E-step to obtain more accurate estimates of the hidden variables. EM theory tells us that in each iteration the estimation of the parameters will be improved and it will eventually converge to a local optimum. In the following we give a more detailed description of how to apply EM theory in our domain.

3.1 E-Step

Let Θ denote the parameters of the graph-based model we want to estimate and $\Theta^{(i-1)}$ denote the estimation thereof at the $i-1$-th iteration of the EM algorithm. The model parameters contain all conditional probabilities needed to describe the dynamic system shown in Fig. 1. The E-step estimates

$$p(x_{1:t} \mid z_{1:t}, \Theta^{(i-1)}), \tag{2}$$

i.e. the posterior distribution over the trajectories of the person given the observations and parameters updated in the previous iteration. Here $x_{1:t}$ and $z_{1:t}$ are the states and observations, respectively. Since it is not possible to find a closed-form solution for the posterior over $x_{1:t}$, we have to resort to an approximate approach [16]. Observe that when we do particle filtering using the motion model with parameter $\Theta^{(i-1)}$, the particle distribution at each time t along with the history of particles is an approximation for $p(x_{1:t} \mid z_{1:t}, \Theta^{(i-1)})$. Hence, the desired expectation can be computed using the graph-based particle filter described in Sect. 2.2. Before we give implementation details for the E-step, let us take a closer look at the M-step.

3.2 M-Step

The goal of the M-step is to maximize the expectation of $\log p(z_{1:t}, x_{1:t} \mid \Theta)$ over the distribution of $x_{1:t}$ obtained in the E-step by updating the parameter estimations. Because the distribution of $x_{1:t}$ is represented by the history of particles, the estimation of the parameters at the i-th EM iteration is computed by summing over all trajectories:

$$\begin{aligned}\Theta^{(i)} &= \underset{\Theta}{\operatorname{argmax}} \sum_{j=1}^{n} \log p(z_{1:t}, x_{1:t}^{(j)} \mid \Theta) \\ &= \underset{\Theta}{\operatorname{argmax}} \sum_{j=1}^{n} (\log p(z_{1:t} \mid x_{1:t}^{(j)}) + \log p(x_{1:t}^{(j)} \mid \Theta)) &(3) \\ &= \underset{\Theta}{\operatorname{argmax}} \sum_{j=1}^{n} \log p(x_{1:t}^{(j)} \mid \Theta) &(4)\end{aligned}$$

Here, n is the number of particles, $x_{1:t}^{(j)}$ is the state history of the j-th particle, and (3) follows from the independence condition

$$p(z_{1:t} \mid x_{1:t}^{(j)}, \Theta) = p(z_{1:t} \mid x_{1:t}^{(j)}),$$

i.e., observations are independent of model transition parameters if the state trajectory is known. For simplicity, we assume that all the particles have equal weight, *i.e.* after they are resampled. It is straightforward to extend our derivation to the case of different weights.

Our approach is in fact a direct extension of the Monte Carlo EM algorithm [17]. The only difference is that we allow particles to evolve with time. It has been shown that when the number of particles n is large enough, Monte Carlo EM estimation converges to the theoretical EM estimation [16].

3.3 Implementation Details

Even though EM can be used to learn all parameters Θ of the model described in Sect. 2, we are mostly interested in learning those parts of the model that describe the typical motion patterns of a user. All other parameters are fixed beforehand and not adjusted to a specific user. An advantage of this approach is that it requires much less training data than learning all parameters at once.

The motion patterns of a specific user are described by the location transitions on the graph and the mode transitions at the different locations. For the learning process, we have to initialize these probabilities to some reasonable values:

$p(e_t \mid e_{t-1}, m_{t-1})$ is the transition probability on the graph conditioned on the mode of transportation just prior to transitioning to the new edge. This conditional probability is initialized to a uniform distribution across all outgoing edges, with the exception of bus routes which have a strong bias forcing buses to follow the route (bus routes can be obtained from GIS sources such as [18]). With this exception, our model has no preference for a specific path of the person.

$p(m_t \mid m_{t-1}, e_{t-1})$ is the mode transition probability. This probability depends on the previous mode m_{t-1} and the location of the person, described by the edge e_{t-1}. For example, each person has typical locations where she gets on and off the bus. Mode transitions are initialized with commonsense knowledge (*e.g.*, one may not switch from a bus to a car without first being on foot), and with knowledge of bus stops. Parking lots are uniformly distributed across our map with no biases toward actual parking lots.

A straightforward implementation of the E-step given in (2) is to generate the expectation over state trajectories by storing the history of each particle (see [7] for a discussion). To do so, at each re-sampling phase, the history of old samples needs to be copied to the new samples[2]. Then at the last time step, we have a set of samples with their histories. At the M-step, we update the model parameters simply by counting over the particle histories. For example, to get $p(e_j \mid e_i, BUS)$, we count the number of times when a particle in BUS mode transits from edge e_i to e_j and then normalize the counts over all edges following e_i and BUS. This approach, although easy to implement, suffers from two drawbacks. First, it is not efficient. When the data log is fairly long, saving the histories for all the particles needs a large amount of space and history replication becomes slow. Second, and more importantly, since the number of samples is finite, the repetition of the re-sampling will gradually diminish the number of different histories and eventually decrease the accuracy of the particle based approximation [7].

We can overcome these problems by observing that we are only interested in learning the discrete transitions between edges and modes, *e.g.*, the probability of transiting from edge e_i to edge e_j in BUS mode. The discreteness of these transitions allows us to apply the well-known Baum-Welch algorithm [15], an EM algorithm for hidden Markov models (HMM). The Monte Carlo version of the Baum-Welch algorithm [19] performs at each iteration both a forward and a backward (in time) particle filtering step. At each forward and backward filtering step, the algorithm counts the number of particles transiting between the different edges and nodes. To obtain probabilities for the different transitions, the counts of the forward and backward pass are normalized and then multiplied at the corresponding time slices.

To show how it works, we define:

$\alpha_t(e_t, m_t)$ is the number of particles on edge e_t and in mode m_t at time t in the *forward* pass of particle filtering.

$\beta_t(e_t, m_t)$ is the number of particles on edge e_t and in mode m_t at time t in the *backward* pass of particle filtering.

$\xi_{t-1}(e_t, e_{t-1}, m_{t-1})$ is the probability of transiting from edge e_{t-1} to e_t at time $t-1$ and in mode m_{t-1}.

[2] Unnecessary copy operations can be avoided by using tree data structures to manage pointers describing the history of particles.

$\psi_{t-1}(m_t, m_{t-1}, e_{t-1})$ is the probability transiting from mode m_{t-1} to m_t on edge e_{t-1} at time $t-1$.

A short derivation gives us [15,19],

$$\xi_{t-1}(e_t, e_{t-1}, m_{t-1}) \propto \alpha_{t-1}(e_{t-1}, m_{t-1}) p(e_t \mid e_{t-1}, m_{t-1}) \beta_t(e_t, m_{t-1}) \quad (5)$$

and

$$\psi_{t-1}(m_t, m_{t-1}, e_{t-1}) \propto \alpha_{t-1}(e_{t-1}, m_{t-1}) p(m_t \mid m_{t-1}, e_{t-1}) \beta_t(e_{t-1}, m_t) \quad (6)$$

After we have ξ_{t-1} and ψ_{t-1} for all the t from 2 to T, we update the parameters as[3]:

$$\begin{aligned} p(e_t \mid e_{t-1}, m_{t-1}) &= \frac{\text{expected number of transitions from } e_{t-1} \text{ to } e_t \text{ in mode } m_{t-1}}{\text{expected number of transitions from } e_{t-1} \text{ in mode } m_{t-1}} \\ &= \frac{\sum_{t=2}^{T} \xi_{t-1}(e_t, e_{t-1}, m_{t-1})}{\sum_{t=2}^{T} \sum_{e_t \in Neighbors\ of\ e_{t-1}} \xi_{t-1}(e_t, e_{t-1}, m_{t-1})} \end{aligned} \quad (7)$$

and similarly

$$\begin{aligned} p(m_t \mid m_{t-1}, e_{t-1}) &= \frac{\text{expected number of transitions from } m_{t-1} \text{ to } m_t \text{ on edge } e_{t-1}}{\text{expected number of transitions from } m_{t-1} \text{ on edge } e_{t-1}} \\ &= \frac{\sum_{t=2}^{T} \psi_{t-1}(m_t, m_{t-1}, e_{t-1})}{\sum_{t=2}^{T} \sum_{m_t \in \{BUS, FOOT, CAR\}} \psi_{t-1}(m_t, m_{t-1}, e_{t-1})} \end{aligned} \quad (8)$$

The complete implementation is depicted in Table 1. As the number of particles increases, the approximation converges to the theoretical EM estimation. Fortunately, our approach is very efficient in this regard, since our model parameters are associated with the number of edges and modes in the graph, not the number of particles.

In addition to the user specific parameters our model requires the specification of other parameters, such as motion velocity and the GPS sensor model. The motion velocity is modeled as a mixture of Gaussians from which velocities are drawn at random. The probabilities of the mixture components depend on the current motion mode and can be learned beforehand using data labeled with the correct mode of motion. We use a standard model to compute the likelihood $p(z_t \mid x_t)$ of a GPS sensor measurement z_t given the location x_t of the person [1].

4 Experiments

Our test data set consists of logs of GPS data collected by one of the authors. The data contains position and velocity information collected at 2-10 second intervals during periods of time in which the author was moving about outdoors. This data was hand labeled with one of three modes of transportation: foot, bus, or car. This labeling was

[3] Usually we also need a prior number for each transition. We will not discuss how to set the prior value in this paper.

Table 1. EM-based parameter learning algorithm.

Model Initialization: Initialize the model parameters $p(e_t|e_{t-1}, m_{t-1})$ and $p(m_t|m_{t-1}, e_{t-1})$.
E-step

1. Generate n uniformly distributed samples and set time $t = 1$.
2. Perform forward particle filtering:
   (a) Sampling: generate n new samples from the existing samples using the current parameter estimation $p(e_t|e_{t-1}, m_{t-1})$ and $p(m_t|m_{t-1}, e_{t-1})$.
   (b) Importance sampling: reweight each sample based on observation z_t.
   (c) Re-sampling: multiply / discard samples according to their importance weights.
   (d) Count and save $\alpha_t(e_t, m_t)$
   (e) Set $t = t + 1$ and repeat (2a)-(2d) until $t = T$.
3. Generate n uniformly distributed samples and set $t = T$.
4. Perform backward particle filtering:
   (a) Compute backward parameters $p(e_{t-1}|e_t, m_t), p(m_{t-1}|m_t, e_t)$ from $p(e_t|e_{t-1}, m_{t-1})$ and $p(m_t|m_{t-1}, e_{t-1})$
   (b) Sampling: generate n new samples from the existing samples using the backward parameter estimation.
   (c) Importance sampling: reweight each sample based on observation z_t.
   (d) Re-sampling: multiply / discard samples according to their importance weights.
   (e) Count and save $\beta_t(e_t, m_t)$
   (f) Set $t = t - 1$ and repeat (4b)-(4e) until $t = 1$.

M-step

1. Compute $\xi_{t-1}(e_t, e_{t-1}, m_{t-1})$ and $\psi_{t-1}(m_t, m_{t-1}, e_{t-1})$ using (5) and (6) and then normalize.
2. Update $p(e_t|e_{t-1}, m_{t-1})$ and $p(m_t|m_{t-1}, e_{t-1})$ using (7) and (8).

Loop Repeat E-step and M-step using updated parameters until model converges.

useful for validating the results of our unsupervised learning, but was not used by the EM learning process.

From this data set, we chose 29 episodes representing a total of 12 hours of logs. This subset consists of all of portions of the data set which were bounded by GPS signal loss, *i.e.* had no intermediate loss of signal of more than 30 seconds, and which contained a change in the mode of transportation at some point in the episode. These episodes were divided chronologically into three groups which formed the sets for three-fold cross-validation for our learning. Fig. 3 shows one of the cross-validation groups used for training. The street map was provided by the US Census Bureau [20] and the locations of the bus stops come from the King County GIS office [18].

4.1 Mode Estimation and Prediction

One of the primary goals of our approach is learning a motion model that predicts transportation routes, conditioned on the mode of transportation. We conducted an experiment to validate our models' ability to correctly learn the mode of transportation at any given instant. For comparison we also trained a decision tree model using supervised learning

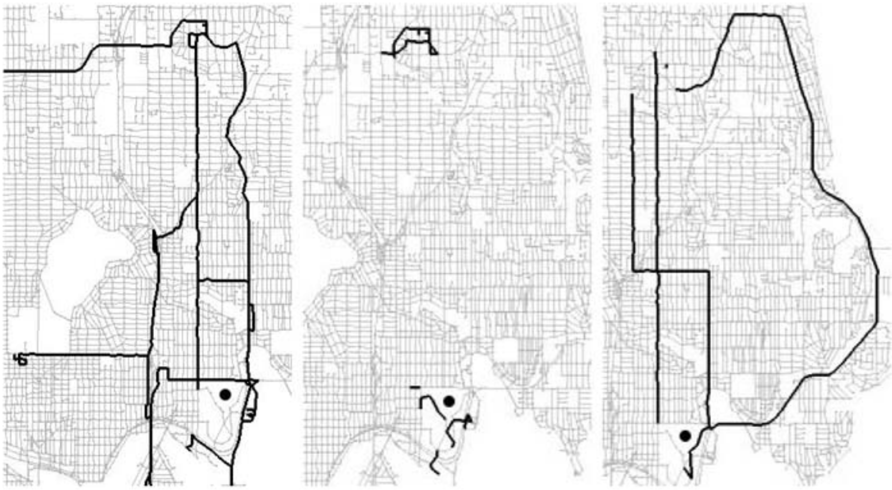


Fig. 3. Car (left), Foot (middle), and Bus (right) training data used for experiments. The black dot is a common map reference point on the University of Washington campus.

Table 2. Mode estimation quality of different algorithms.

Model	Cross-Validation Prediction Accuracy
Decision Tree with Speed and Variance	55%
Prior Graph Model, w/o bus stops and bus routes	60%
Prior Graph Model, w/ bus stops and bus routes	78%
Learned Graph Model	84%

on the data [21]. We provided the decision tree with two features: the current velocity and the standard deviation of the velocity in the previous sixty seconds. Using the data annotated with the hand-labeled mode of transportation, the task of the decision tree was to output the transportation mode based on the velocity information. We used three-fold cross-validation groups to evaluate all of the learning algorithm. The results are summarized in the first row of Table 2. The first result indicates that 55% of the time the decision tree approach was able to accurately estimate the current mode of transportation on the test data. Next, we used our Bayes filter approach without learning the model parameters, *i.e.* with uniform transition probabilities. Furthermore, this model did not consider the locations of bus stops or bus routes (we never provided parking locations to the algorithm). In contrast to the decision tree, the Bayes filter algorithm integrates information over time, thereby increasing the accuracy to 60%. The benefit of additionally considering bus stops and bus routes becomes obvious in the next row, which shows a mode accuracy of 78%. Finally, using EM to learn the model parameters increases the accuracy to 84% of the time, on test data not used for training. Note that this value is very high given the fact that often a change of transportation mode cannot be detected instantaneously.

Table 3. Prediction accuracy of mode transition changes.

Model	Precision	Recall
Decision Tree with Speed and Variance	2%	83%
Prior Graph Model, w/o bus stops and bus routes	6%	63%
Prior Graph Model, w/ bus stops and bus routes	10%	80%
Learned Graph Model	40%	80%

A similar comparison can be done looking at the techniques' ability to predict not just instantaneous modes of transportation, but also *transitions* between transportation modes. Table 3 shows each technique's accuracy in predicting the qualitative change in transportation mode within 60 seconds of the actual transition — for example, correctly predicting that the person got off a bus. Precision is the percentage of time when the algorithm predicts a transition that an actual transition occurred. Recall is the percentage of real transitions that were correctly predicted. Again, the table clearly indicates the superior performance of our learned model. Learning the user's motion patterns significantly increases the precision of mode transitions, *i.e.* the model is much more accurate at predicting transitions that will actually occur.

An example of the modes of transportation predicted after training on one cross-validation set is shown in Fig. 4.

4.2 Location Prediction

The location prediction capabilities of our approach are illustrated in Fig. 5 and 6. In Fig. 5, the learned model was used to predict the location of the person into the future. This was done by providing the ground truth location and transportation mode to the algorithm and then predicting the most likely path based on the transition probabilities learned from the training data. The figure shows the percentage of trajectories that were predicted correctly, given different prediction horizons. Prediction length was measured in city blocks. For example, in 50% of the cases, the location of the person was predicted correctly for 17 blocks when the person was on the bus. In 30% of the cases, the prediction was correct for 37 blocks, and 75 blocks were predicted correctly in 10% of the cases. Note that the linear drop in bus route prediction probability is due to the fact that the data contained several correctly predicted episodes of a 92 block long bus trip. Obviously, long term distance prediction is much less accurate when a person walks. This is due to the higher variability of walking patterns and the fact that people typically do not walk for many city blocks, thereby making a long term prediction impossible.

In Fig. 6, the learned model was used to predict both the location and the transportation mode of the person into the future. This was done by providing the ground truth location to the algorithm and then predicting the most likely path and sequence of transportation mode switches based on the transition probabilities learned from the training data. The graph shows that in 50% of the cases, the model is able to correctly predict the motion and transportation mode of the person for five city blocks. This result is extremely promising given that the model was trained and tested on subsets of 29 episodes.

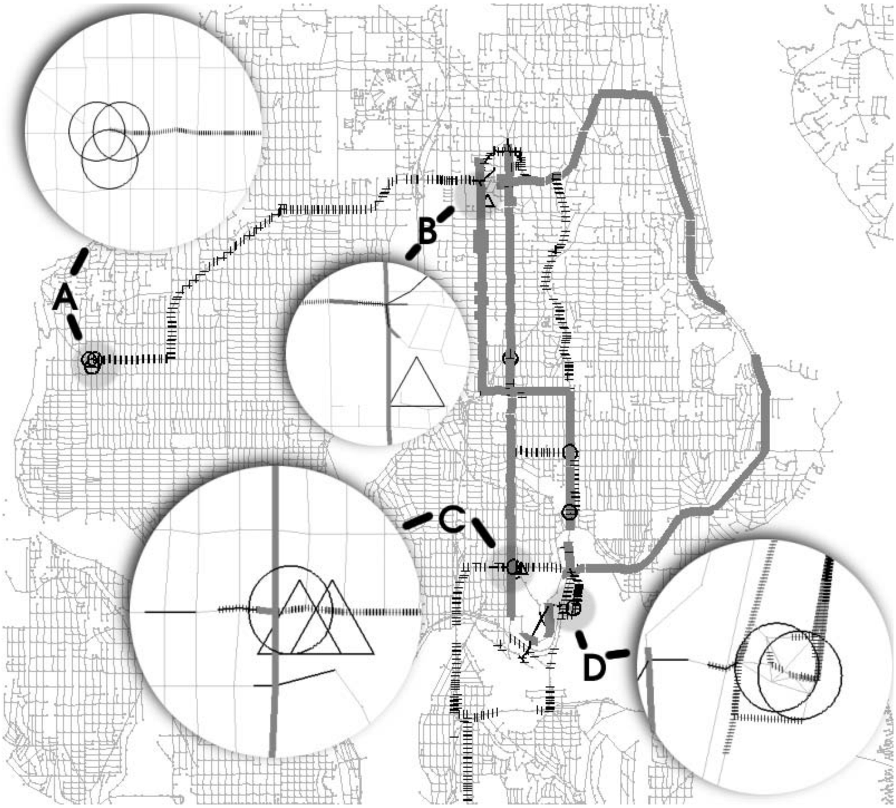


Fig. 4. This map shows the learned transportation behavior based on one cross-validation set containing nineteen episodes. Shown are only those edges and mode transitions which the learned model predicts with high probabilities. Thick gray lines indicate learned bus routes, thin black lines indicate learned walking routes, and cross-hatches indicate learned driving routes. Circles indicate parking spots, and the triangles show the subset of bus stops for which the model learned a high probability transition on or off the bus. There are four call-outs to show detail. **(A)** shows a frequently traveled road ending in three distinct parking spaces. This route and the parking spots indicate the correctly learned car trips between the author's home and church. **(B)** shows a frequently traveled foot route which enters from the northeast, ending at one of the frequently used bus stops of the author. The main road running east-west is an arterial road providing access to the highway for the author. **(C)** shows an intersection at the northwest of the University of Washington campus. There are two learned bus stops. The author frequently takes the bus north and south from this location. This is also a frequent car drop off point for the author, hence the parking spot indication. Walking routes extend west to a shopping area and east to the campus. **(D)** shows a major university parking lot. Foot traffic walks west toward campus.

5 Conclusions and Future Work

The work presented in this paper helps lay the foundation for reasoning about high-level descriptions of human behavior using sensor data. We showed how complex behaviors

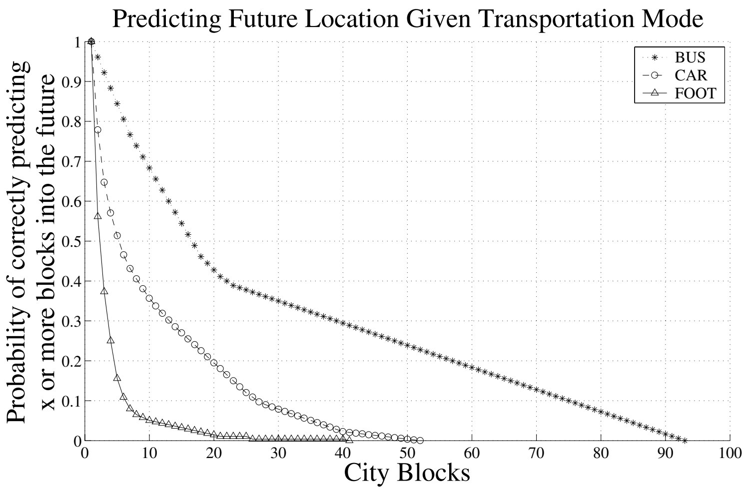


Fig. 5. Location prediction capabilities of the learned model.

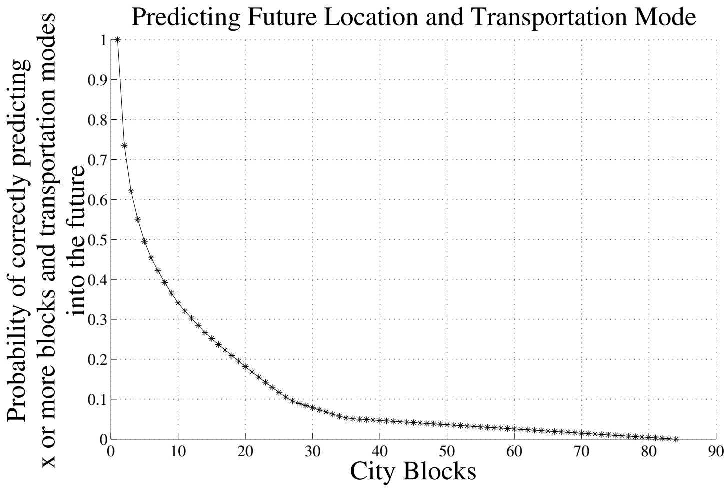


Fig. 6. Location and mode prediction capabilities of the learned model.

such as boarding a bus at a particular bus stop, traveling, and disembarking can be recognized using GPS data and general commonsense knowledge, without requiring additional sensors to be installed throughout the environment. We demonstrated that good predictive user-specific models can be learned in an unsupervised fashion.

The key idea of our approach is to apply a graph-based Bayes filter to track a person's location and transportation mode on a street map annotated with bus route information. The location and transportation mode of the person is estimated using a particle filter. We showed how the EM algorithm along with frequency counts from the particle filter can be used to learn a motion model of the user. A main advantage of this unsupervised learning algorithm is the fact that it can be applied to raw GPS sensor data.

The combination of general knowledge and unsupervised learning enables a broad range of "self-customizing" applications, such as the Activity Compass mentioned in Sect. 1. Furthermore, it is straightforward to adopt this approach for "life long" learning: the user never needs to explicitly instruct the device, yet the longer the user carries the device the more accurate its user model becomes.

Our current and future research extends the work described in this paper in a number of directions, including the following:

1. *Making positive use of negative information.* Loss of GPS signal during tracking causes the probability mass to spread out as governed by the transition model. We have seen that learning significantly reduces the rate of spread. In some cases, however, loss of signal can actually be used to tighten the estimation of the user's location. In particular, most buildings and certain outdoor regions are GPS dead-zones. If signal is lost when entering such an area, and then remains lost for a significant period of time while the GPS device is active, then one can strengthen the probability that the user has not left the dead-zone area.
2. *Learning daily and weekly patterns.* Our current model makes no use of absolute temporal information, such as the time of day or the day of the week. Including such variables in our model will improve tracking and prediction of many kinds of common life patterns, such as the fact that the user travels towards his place of work on weekday mornings.
3. *Modeling trip destination and purpose.* The work described in this paper segments movement in terms of transitions at intersections and between modes of transportation. At a higher level of abstraction, however, movement can be segmented in terms of trips that progress from a location where some set of activities take place (such as home) to a location where a different class of activities take place (such as the office). A single trip between activity centers can involve several shifts between modes of transportation. By learning trip models we expect to be able to increase the accuracy of predictions. More significantly, trip models provide a way to integrate other sources of high-level knowledge, such as a user's appointment calendar.
4. *Using relational models to make predictions about novel events.* A significant limitation of our current approach is that useful predictions cannot be made when the user is in a location where she has never been before. However, recent work on relational probabilistic models [22,23] develops a promising approach where predictions can be made in novel states by smoothing statistics from semantically similar states. For example, such a model might predict that the user has a significant chance of entering a nearby restaurant at noon even if there is no history of the user patronizing that particular restaurant.

Acknowledgment

This work has partly been supported by the NSF under grant numbers IIS-0225774 and IIS-0093406, by AFRL contract F30602–01–C–0219 (DARPA's MICA program), and the Intel corporation.

References

1. Hightower, J., Borriello, G.: Location systems for ubiquitous computing. Computer **34** (2001) IEEE Computer Society Press.
2. Bonnifait, P., Bouron, P., Crubillé, P., Meizel, D.: Data fusion of four ABS sensors and GPS for an enhanced localization of car-like vehicles. In: Proc. of the IEEE International Conference on Robotics & Automation. (2001)
3. Cui, Y., Ge, S.: Autonomous vehicle positioning with GPS in urban canyon environments. In: Proc. of the IEEE International Conference on Robotics & Automation. (2001)
4. Ashbrook, D., Starner, T.: Learning significant locations and predicting user movement with gps. In: International Symposium on Wearable Computing, Seattle, WA (2002)
5. Patterson, D., Etzioni, O., Fox, D., Kautz, H.: The Activity Compass. In: Proceedings of UBICOG '02:First International Workshop on Ubiquitous Computing for Cognitive Aids. (2002)
6. Kautz, H., Arnstein, L., Borriello, G., Etzioni, O., Fox, D.: The Assisted Cognition Project. In: Proceedings of UbiCog '02: First International Workshop on Ubiquitous Computing for Cognitive Aids, Gothenberg, Sweden (2002)
7. Doucet, A., de Freitas, N., Gordon, N., eds.: Sequential Monte Carlo in Practice. Springer-Verlag, New York (2001)
8. Liao, L., Fox, D., Hightower, J., Kautz, H., Schulz, D.: Voronoi tracking: Location estimation using sparse and noisy sensor data. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems. (2003)
9. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: Estimation with Applications to Tracking and Navigation. John Wiley (2001)
10. Dean, T., Kanazawa, K.: Probabilistic temporal reasoning. In: Proc. of the National Conference on Artificial Intelligence. (1988)
11. Murphy, K.: Dynamic Bayesian Networks: Representation, Inference and Learning. PhD thesis, UC Berkeley, Computer Science Division (2002)
12. Bar-Shalom, Y., Li, X.R.: Multitarget-Multisensor Tracking: Principles and Techniques. Yaakov Bar-Shalom (1995)
13. Del Moral, P., Miclo, L.: Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non linear filtering. In: Seminaire de Probabilites XXXIV. Number 1729 in Lecture Notes in Mathematics. Springer-Verlag (2000)
14. Bilmes, J.: A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report ICSI-TR-97-021, University of Berkeley (1998)
15. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. In: Proceedings of the IEEE, IEEE (1989) IEEE Log Number 8825949.
16. Levine, R., Casella, G.: Implementations of the Monte Carlo EM algorithm. Journal of Computational and Graphical Statistics **10** (2001)
17. Wei, G., Tanner, M.: A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. Journal of the American Statistical Association **85** (1990)
18. County, K.: Gis (graphical information system). http://www.metrokc.gov/gis/mission.htm (2003)
19. Thrun, S., Langford, J., Fox, D.: Monte Carlo hidden Markov models: Learning non-parametric models of partially observable stochastic processes. In: Proc. of the International Conference on Machine Learning. (1999)
20. Bureau, U.C.: Census 2000 tiger/line data. http://www.esri.com/data/download/census2000-tigerline/ (2000)
21. Mitchell, T.: Machine Learning. McGraw-Hill (1997)
22. Anderson, C., Domingos, P., Weld, D.: Relational markov models and their application to adaptive web navigation. In: Proceedings of the Eighth International Conference on Knowledge Discovery and Data Mining, ACM Press (2002) 143–152 Edmonton, Canada.
23. Sanghai, S., Domingos, P., Weld, D.: Dynamic probabilistic relational models. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann (2003) Acapulco, Mexico.

Activity Zones for Context-Aware Computing

Kimberle Koile[1], Konrad Tollmar[2], David Demirdjian[2],
Howard Shrobe[1], and Trevor Darrell[2]

[1] Agent-Based Intelligent Reactive Environments Group
MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139
{kkoile,hes}@ai.mit.edu
[2] Visual Interface Group
MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139
{konrad,demirdji,trevor}@ai.mit.edu

Abstract. Location is a primary cue in many context-aware computing systems, and is often represented as a global coordinate, room number, or a set of Euclidean distances to various landmarks. A user's concept of location, however, is often defined in terms of regions in which similar activities occur. We discuss the concept of such regions, which we call activity zones, and suggest that such zones can be used to trigger application actions, retrieve information based on previous context, and present information to users. We show how to semi-automatically partition a space into activity zones based on patterns of observed user location and motion. We describe our system and two implemented example applications whose behavior is controlled by users' entry, exit, and presence in the zones.

1 Introduction

The utility of an application can be increased greatly by taking into account the specific context in which the application is used. Users, however, do not like the burden of explicitly stating context information. The implicit control of applications using passively sensed context cues frees users from the burden of explicitly specifying service details, e.g., for whom a service should be performed, where it should occur, and how it should be delivered. Location is one of the primary context cues in context-aware systems, e.g., [6]. Many of these systems define location in geometric terms such as a 2D or 3D position coordinate, room identifier, or a set of distances to known landmarks. While these definitions are useful, they neglect how people actually use physical space.

One way to understand the use of physical space is in terms of zones that are created by elements of physical form and that support human activity, e.g., [21, 25]. Architects, for example, design eating places, gathering places, and places of repose, e.g. [3, 24]. Anthropologists and sociologists talk of "hidden zones" in offices [17]. CSCW researchers interested in home environments have studied public and private zones for use of communication media [27], and have identified activity centers where particular tasks take place [10]. In workspaces, we think of zones for quiet, solitary work; zones for informal meetings; zones for formal presentations [14, 34]. A table and accompanying chairs create a zone in which we might meet with colleagues. A whiteboard creates a zone in front of it; people draw on the whiteboard while standing in this "whiteboard zone".

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 90–106, 2003.

We make two key observations about identifying such zones. Thinking bottom up, we can identify zones by observing people as they go about their daily activities, partitioning a space based on people's locations and motions. Thinking top down, we can define a taxonomy of zones based on prototypical human activity and physical form that supports that activity. Such zones, identified using either or both of these approaches, can be thought of as representing regions of similar context. With such activity-dependent zones, one can build more useful context-aware computing applications by (1) identifying meaningful zones to which users can attach semantics and preferred application behavior, e.g., behavior that would be triggered upon entry, exit or presence in a zone; and (2) enabling inference of human activity at various levels of abstraction, e.g., three people sitting in chairs around a table, vs three people having a meeting.

In this paper we describe our implementation of a system that identifies "activity zones" semi-automatically using techniques from computer vision and artificial intelligence. We present an implemented experimental system in which transitions between zones successfully control device and application behavior.

We begin by reviewing previous work, then discuss the concept of activity zones, our experimental system, and ongoing and future research efforts.

2 Previous Work

The study of context and its role in ubiquitous computing systems is an active research field, with many definitions for context and context-awareness, e.g., [12, 13, 32]. Central to the notion of context is location, since many applications and services are conditioned on the place where they should be performed or displayed. Location-aware computing has become a topic of active research, and many schemes have been proposed for providing location cues using IR, RF, ultrasound, and computer vision tracking systems. (For a survey, see [22].)

Early systems employed ad-hoc representations of location, usually tailored to specific sensor data representations. Recently a general scheme for device independent location representation and sensor fusion has been proposed, using a layered abstraction model based on proximity and position measurements [23]. The majority of location-awareness schemes report raw 2D or 3D position information, room identity, and/or proximity to a beacon. These cues are useful for many tasks, but they are indifferent to the physical form or use of the space, and thus are insufficient in some cases. A person may be equally close to a whiteboard or a table, for example. If a display system knows that the person is moving back and forth in a standing position, it can infer that the person is at the whiteboard and that a nearby wall is a better display location than a computer monitor on the table.

A few systems for location awareness are able to report information about subregions or furniture, and/or adapt over time based on observed behavior. The EasyLiving system used a map of regions to indicate places in the environment associated with specific context features, usually furniture [7]. These maps were drawn manually and provided to the system, rather than being learned from observed behavior. Similarly the Sentient Computing System's notion of spatial containment allows bounds for 2D regions associated with positions of active devices [2], e.g., an oval area in front of a computer display. A system for automatically mapping an environment based on the movement of personnel in a location-aware environment was described in [19]. This system was adaptive and learned from observing user behavior, but formed a map of a large scale environment and did not find contextually relevant regions within rooms.

In our current research, we represent context by means of regions that we call activity zones, which are identified by observing users' locations and motions. The term "activity zones" is used in [30] to describe spatial distribution of activities in an office setting. The zones represent regions within a room, as ours do, but are computed by analyzing a large corpus of camera images rather than by means of real-time tracking software. In addition, the zones are used to inform the design of new office layouts rather than in ubiquitous computing applications.

In contrast to previous work, our activity zone representation is both fine-grained (i.e., smaller than a room) and learned from observing patterns of user behavior. It supports applications that can make productive use of inferences about a user's activity beyond simple position and proximity cues, without requiring the user to draw a map. We develop our scheme in an interactive framework: the system learns the geometry of the activity zones automatically from observed behavior, and relies on the user to associate semantics or rules with events related to those zones.

3 Activity Zones

Physical form – e.g., walls, furniture – partitions space into zones that are places of human activity. Walls create a zone in which people might play music; furniture creates a zone in which people might read or talk. In the floor plan shown in Fig. 1, there is a zone created by the doorway, a zone created by the sofa, and a zone created by the corner desk, table, and chair.

Zones defined by physical form are useful as representations of context, but only partially capture a person's context – they ignore a person's use of a space. Rather than building a model of a space and its furniture, as for example in [8], we can partition a space into zones by observing human activities. We call these partitions activity zones; e.g., the area in which a group of people are standing in a hall talking. If we take "observing human activity" to mean recording people's locations and motions, then for a floor plan such as the one shown in Fig. 1, we would ideally find the zones shown in Fig. 2: zone 1 is a region in which people stand, zones 2 and 3 are regions in which people sit, zone 4 is a region in which people walk. Note that zones 1 through 3 correspond to the physical zones shown in Fig. 1. They contain extra information, however – whether the person sits or stands in these zones. Zone 4 corresponds to an access corridor used for walking between the three other zones. To identify such a zone in a model that represents only the physical space, such as that shown in Fig. 1, one would have to represent circulation paths explicitly.

Both physical zones, shown in Fig. 1, and activity zones, shown in Fig. 2, can be thought of as representing regions of similar context. Physical zones represent location contexts; activity zones represent location and motion contexts. Physical zones could be inferred from observation of static furniture configuration, but activity zones need to be learned from statistics of human behavior. Application behaviors can be controlled by a person's entry, exit, or presence in either type of zone. Activity zones, with the extra information about user motion, enable application behaviors to be tied more closely to what people are doing rather than just where they are.

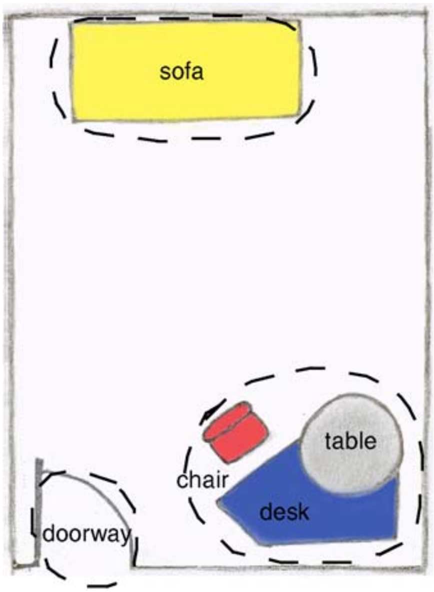


Fig. 1. Three zones created by physical form

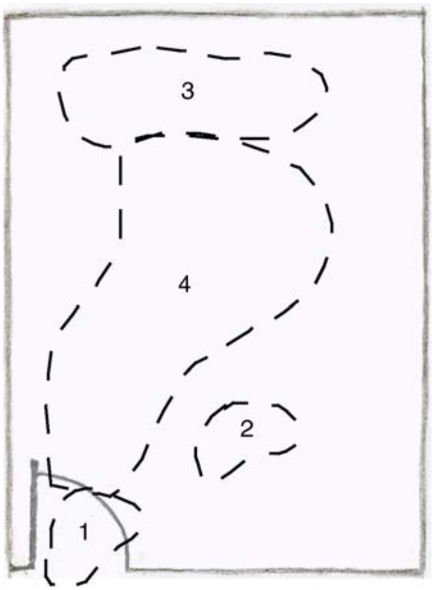


Fig. 2. Four zones identified by observing human activities

To construct activity zones, a tracking system observes people's activities over time, say a week, and constructs a map of activity zones. A user, or eventually a machine learning program, then may attach preferred application behavior to activity zones. A user also might attach semantics to the activity zones so that application behaviors could be specified for types of zones instead of individual zones. If a user, for example, did not want to receive phone calls while reading, she might label a particular zone as a reading zone, and indicate that calls should be held while she is reading. Once preferred behavior has been specified, the tracking system posts events about people's entry, exit, or presence in particular zones. An accompanying notification system informs interested applications of the events, and the applications react accordingly.

In this description of activity zones, the map is static – it is created by observing people in a space, then used assuming that the arrangement of space does not change. Yet zones are often correlated with furniture location, and in today's workspaces furniture is often moved. What happens, then, when a sitting zone no longer contains the chair it once did? One could map furniture locations to zone locations when an activity zone map is created, then periodically check that the current furniture locations match those in the current zone map. We discuss the issue of dynamic activity zones further in Sect. 7.

4 Scenarios

The following scenarios illustrate the activity zones concept. Consider a workspace containing zones such as those shown in Figs. 3 and 4. Note that the zone near the round moveable table in Fig. 4 is what we have called a dynamic activity zone.

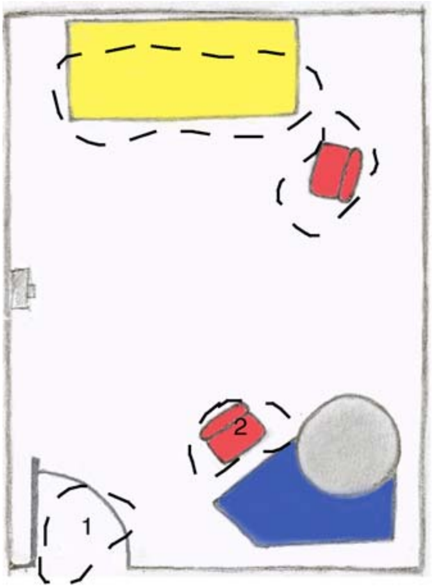


Fig. 3. Consider two zones, one at doorway one at corner desk chair; projector is shown at left

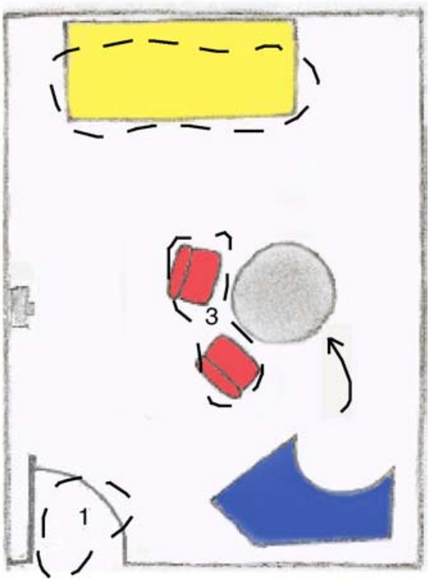


Fig. 4. New zone is created when round table is moved

Scenario A: Jane walks into her office and the overhead lights and the teapot turn on; the newspaper headlines are displayed on the wall near her desk. The room greets her and asks if she would like to be reminded of her calendar events for the day. She says yes, sits down at her desk, and her calendar is displayed on the computer screen at her desk. She notices that a student, Lauren, is coming by soon for a meeting about a term project. Jane asks the room to retrieve the notes from her last meeting with Lauren and to project them when she and Lauren start their meeting.

Scenario B: Lauren arrives, Jane greets her and invites her to come in. Jane moves the table into the middle of the room and invites Lauren to sit down at the table. As Lauren sits down, the notes about her project are projected on the wall between the table and desk. Jane and Lauren start to discuss the project.

Scenario C: After discussing the project, Lauren leaves. As Jane moves the table back to its original location, the projector turns off. Back at her desk, Jane notices that the message light on her phone is lit and asks the room to play her phone messages.

In Scenario A, several preferred behaviors were triggered upon entry into the doorway zone: "turn on overhead lights", "display news headlines", "turn on teapot", "ask about showing calendar". Jane's entry into the zone near the desk triggered display of the calendar on the display device appropriate for that zone, the computer screen on the desk. Scenario A also illustrates the creation of a dynamic event trigger: Jane requests that the room do something when a future event occurs, namely that it display notes when her meeting starts. Note that in order to identify a meeting, the room would need additional knowledge. It would need to know, for example, that meetings happen when more than one person is in zone 3, or that meetings happen at tables and there is a table in zone 3. Without this additional knowledge, Jane would have had to request that the room display the information when she and Lauren entered zone 3.

Scenario B illustrates the display of information in an appropriate place based on the meeting starting in zone 3. Simple proximity to devices would not have worked in this example, because Jane and Lauren were equidistant between the projector and the computer display on the corner desk.

Scenario C illustrates that phone calls are held during meetings. It also shows the projector turning off when Jane moves the table back to its original location.

Together these scenarios illustrate the use of context to trigger room actions automatically, to retrieve information, and to present that information to users. These uses of context are similar to those discussed in [8] and [13].

5 Implementation

We have implemented an activity zone system and two of the context-aware application behaviors mentioned in the above scenarios: device control and selection of display location and method. Our activity zone system is part of a larger system that provides services in an intelligent environment. It embodies a perceive-reason-act paradigm, as illustrated in Fig. 5, and is organized using a blackboard architecture [15]. Perceptual systems, such as the person tracker, post events to a blackboard. The blackboard provides a reasoning system with a shared memory that represents the current state of the world – e.g., the activity zone map in use; pending and processed events; the people in the space and their contexts, represented as motions and entry, exit, or presence in particular zones. Knowledge sources associated with the blackboard do forward inference, abstracting events into higher level context statements (e.g., a meeting in zone 3) or mapping events to requests for action (e.g., turn on the lights). The requests for action are sent to device controllers and other applications, which in turn process the requests.

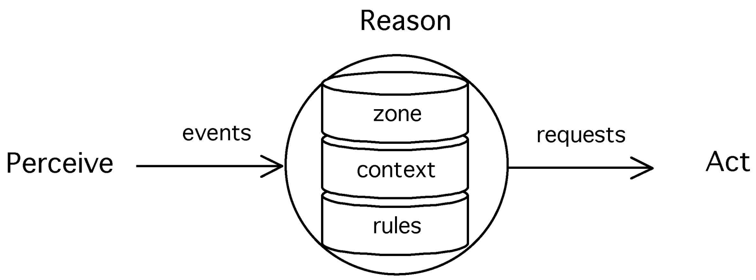


Fig. 5. Overview of system architecture

We have installed our system in a workspace similar to the one described in the above scenarios, focusing on furniture that can be configured easily to create zones for individual work and collaborative work. Fig. 6 is a sketch of the floor plan. Fig. 7 is a photograph of the space.

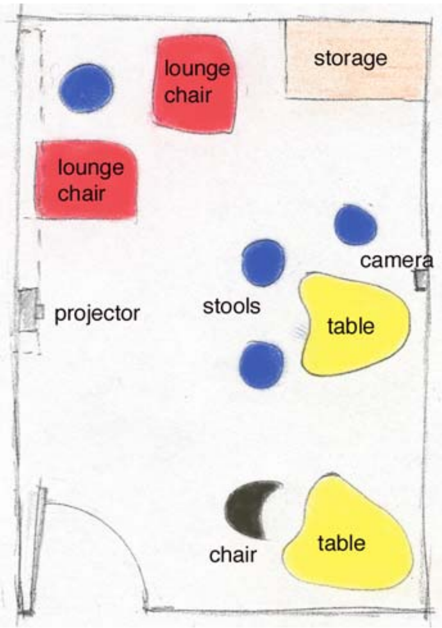


Fig. 6. Floor plan sketch of workspace

Fig. 7. View of workspace from doorway

5.1 Tracking System

To support activity zones, a person tracking system must provide information in real-time on the number of people in a space and each person's location and motion, e.g., represented by height and velocity. These are general requirements, and many different tracking approaches can be used, including those that track wearable or handheld devices using IR, RF, infrared, GPS, 802.11 [22]. We have chosen to use a passive person tracking system based on computer vision because it does not require any additional devices to be worn or carried by users. The concept of activity zones and our techniques for constructing and manipulating zones, however, are independent of the tracking system used, as long as the tracker provides the information specified above.

We use a multi-camera stereo-based tracking system to track people in indoor environments. (More details about the tracking system can be found in [11].) The tracker provides a history of 3D information for every person in the observed space. The information is a triple (x, y, h), where x, y are the coordinates of the person in the ground plane, and h is the height of the top of her head above the floor. Since tracking data are time-stamped, the instantaneous velocity (v_x, v_y, v_h) can be derived. We then characterize a person at location (x, y) using the activity feature $f(x,y) = (h, v, v_{lt})$, where h is the height, v is the instantaneous ground plane velocity norm, and v_{lt} is the average ground plane velocity norm over a certain period of time. By using the activity feature $f(x, y)$, we can capture the configuration (sitting, standing) and movement of a person over both short and long periods of time.

To estimate an activity zone map, we track people in a space for a period of time, collecting a dense set of activity features $f(x, y)$ and locations (x, y), then segment the activity features using a two-step clustering process. We first cluster the activity features into classes, each representing a similar activity (i.e., configuration and movement), which is represented as an average activity feature F_k. Then for each class, we cluster the associated (x, y) locations into regions. The resulting regions represent activity zones, regions in 2D space that are characterized by an average activity F_k. As different activities may happen at the same location, activity zones may overlap. Once an activity zone map has been created, a person's entry, exit, or presence in an activity zone is identified by matching the person's instantaneous location and activity features to the map. (For more details of the clustering and matching algorithms, see [11].)

The length of time for collecting data for an activity zone map is an open research question; we typically track people for a day. People can be detected with an accuracy of about 20 to 30 centimeters. We have run experiments with the system successfully tracking 8 people in the space at the same time.

The figure below shows a portion of an activity map for our workspace. The map is represented using simple XML primitives in order to allow heterogeneous agents, such as one that provides a graphical user interface for visualizing and labeling zones, to easily access and manipulate the map data. In this example, the map contains three zones numbered 0, 1, 2 and labeled "desk", "table", and " lounge". The numbers are supplied by the tracker; labels are added by the user via a simple graphical user interface. Displays from that interface, which we call the zone editor, are shown in Figs. 9 and 10.

```
<?xml version="1.0"?>
 <amap xsize=200 ysize=200 im="835-PTGO.jpg">

   <zone id=0 label="desk" height=1.1 velocity=0.1 color="ff0000">
        12,182 12,184 38,188 …
   </zone>

   <zone id=1 label="table" height=1.7 velocity=0.5 color="00ff00">
        100,182 100,180 89,154 …
   </zone>

   <zone id=2 label="lounge" height=0.8 velocity=0.05 color="0000ff">
      150,130 150,132 135,118 …
   </zone>

  </amap>
```

Fig. 8. Activity-map in XML. Labels are user supplied

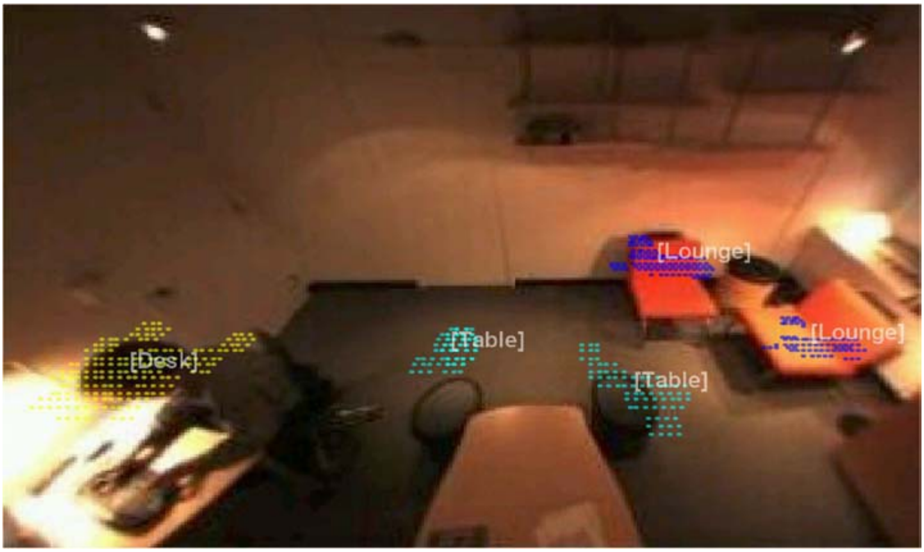


Fig. 9. The workspace from the camera's perspective, overlaid with clustered tracker data for three zones; labels (desk, table, lounge) are user-supplied

Fig. 9 shows the 2D extent for each of three activity zones clustered around furniture groupings. The tracker found a total of 11 zones. Using the zone editor, the user aggregated smaller zones into three zones of interest and pruned others. The table zone, for example, is an aggregation of zones around each of the stools. An access zone through the middle of the room was pruned. Fig. 10 shows an alternate zone map.

Fig. 10. The workspace overlaid with clustered tracker data for three different zones; zone labels (door, desk, lounge) are user-supplied

5.2 Blackboard and Infrastructure

Our perceive-reason-act paradigm is centered around a blackboard, which provides a shared memory through which system components communicate. (See Fig. 11.) The blackboard contains a context memory, a set of zone maps, a current zone map, a perceptual event queue, and a requested action queue. We use an implementation similar to that described in [26] and [33][1]. Perceptual systems, such as the tracker, post events to the blackboard. An agent-based system [9,18] provides the communication layer between the perceptual systems and the blackboard. By means of a publish-and-subscribe mechanism, the blackboard registers interest in particular classes of notifications, e.g., tracker events. Incoming events are added to the event queue. Once events are posted to the blackboard, a context inference system then reasons forward from the events, using such rules as "if there is more than 1 person in a zone then there is a meeting in the zone". Resulting inferences, such as "person 1 in a meeting", are posted to the blackboard as assertions in the context memory. In the terminology of a traditional blackboard model, the context inference system is represented as a set of knowledge sources, each of which is a set of rules contributing inferences at higher levels of abstraction than "raw" tracker events. These higher levels of abstraction allow users to specify preferences more naturally, e.g., in terms of a meeting rather than in terms of the number of people in a zone. They also enable preferences to work in the presence of dynamic activity zones since users can associ-

[1] Event queues can be inefficient because they force sequential processing of events. We have not found this to be a problem because most of our events thus far have come from a single stream of tracker data. See [26] for discussion of a principled approach to avoiding slowdowns due to event queues.

ate preferences with types of zones, e.g., meeting zones, rather than particular zones anchored in physical space.

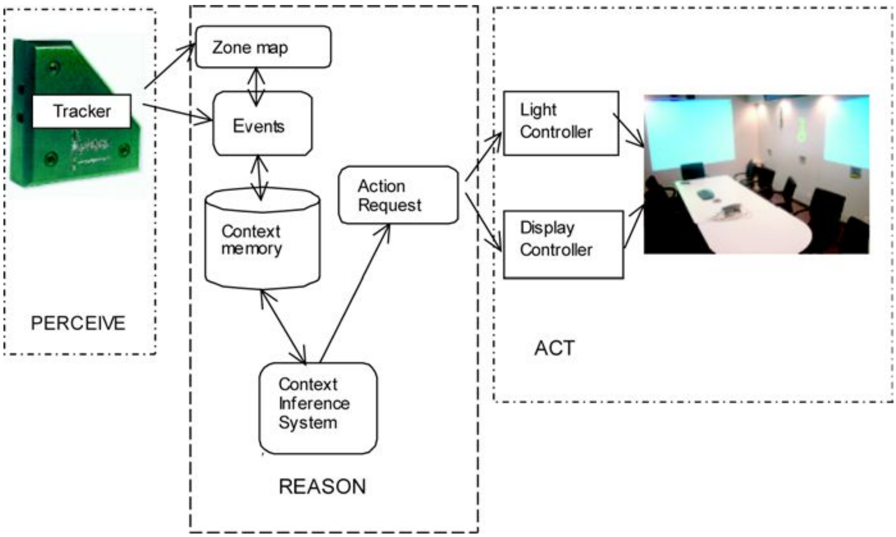


Fig. 11. System architecture

In addition to posting context assertions, the context inference system may post requests for action on an action request queue. In our current implementation, requests are processed sequentially by a central controller that aggregates similar requests and resolves conflicts between dissimilar requests. It, for example, will ignore a "turn off" request that is immediately followed by a "turn on" request for the same light. We currently are investigating the tradeoffs between centralized and decentralized control of device and application behavior.

5.3 User Interaction

As previously mentioned, the system is semi-automatic. It automatically determines zones and carries out preferred device and application behaviors. It relies on the user to: select and label zones of interest, add device and application behavior preferences to the zones, and define high level contexts, e.g., what constitutes a meeting. User interaction is via a simple graphical user interface, which we call the zone editor, that uses the displays shown in Figs. 9 and 10. Preferred behaviors are added to zones using a form template and a simple rule-like language. Examples of the language are shown in Fig. 12. Examples of specifying high level contexts, such as meetings, are shown in the next section.

```
if enter zone 0
then turn-on desk-lamp

if enter zone lounge
then turn-on lounge-lamp
     and turn-off desk-lamp
```

Fig. 12. Examples of specifying preferred device behavior

6 Example Applications

We describe two implemented examples of using activity zones as context cues. The first example, control of lights and audio, uses a direct mapping from events to environment actions. The second example, delivering a message based on context, demonstrates the use of inference to aggregate zones and to hypothesize high level context descriptions.

6.1 Device Control

The system adjusts light and audio levels when people enter and exit particular activity zones. We used our tracking system to generate activity zones as described in the previous section. When we examined the resulting activity zone map, we discovered a small zone around each of the stools. We found it more useful to think of the stools as a single zone, especially when it came to specifying preferred device behavior. We aggregated the zones into a table zone using the zone editor mentioned in Sects. 5.1 and 5.3.

Typical device settings were:

- When someone enters the desk zone, turn on the lamp and music
- When someone enters the lounge zone, turn on the lamp in that zone and turn off the lamp in the desk zone
- When a second person enters the table zone, turn on the projector and turn off the music

When a person moved around in the room, tracker events for entry to and exit from zones were translated into requests for device control by means of the preference settings. Fig. 13 shows the state of lights and projector with two people sitting in the table zone.

We informally observed people working at the desk and meeting with colleagues at the table. We noted that the activity zones correlated well with particular activities, e.g., typing at the keyboard in the desk zone, reading and talking with others in the lounge zone, working alone or with others in the table zone. People described the zones by either the objects in the zone or by the activities that took place in the zones. Most would have preferred that zones be labeled automatically rather than by hand.

Object recognition and activity inference could be used to provide such labels and to insure that labels carry semantic information rather than being just symbols. Informally, we found that people working in the room liked the organization of the room and thought the automatic control of lights, radio, and projector novel and useful. (A forthcoming memo will describe studies currently underway.)

Fig. 13. Two people in the table zone. Lamp near desk shown at right has turned off; projector has turned on

6.2 Message Delivery

We implemented a message delivery system for context-aware notification in order to test inference of high level contexts, such as having a meeting, and for specification of delivery preferences. Examples of context inference and delivery preference rules are shown in Fig. 14. We also used inference rules for aggregation of several zones into a single larger zone as an alternative to hand editing of zones. Aggregation using rules works as follows. If we assume, for example, that we care about a zone C (e.g., near a table), that is the union of zones A and B (e.g., around two stools), then the inference system represents the aggregation of A and B using a rule such as "if a person enters zone A or zone B then the person enters zone C". In essence, we use the rules to create a conceptual zone map that aggregates smaller zones into larger ones. When the tracker posts an event to the blackboard that a person has entered zone A, the inference system sees that event and infers that the person has entered zone C. If there are preferred application behaviors attached to entry in zone C, the inference system then can post the appropriate requests for action.

In our implemented example, we used rules such as those shown in Fig. 14. Rules 1 and 2 aggregate two stool zones into a table zone. Rules 3 and 4 specify that a person is in a meeting when the zone she is in contains at least one other person. Rules 5 and 6 specify message delivery preferences.

We had two people sit on the stools, and we sent a message to one of them, which caused a message delivery event to be posted to the blackboard[2]. The tracker noticed a person in each of the two stool zones; the context inference system inferred that

[2] People were identified by the system with an explicit utterance.

each person was in the table zone and that there was a meeting in that zone. The message then was delivered without being displayed – it appeared as an icon on the computer screen in the desk zone, which was the default display device for the recipient when she was in a meeting. With one person in the room, a message is displayed on the computer screen if the person is in the desk zone (where the screen is), or projected on the wall if the person is in the lounge zone. Anecdotally, experiences of novice users (computer science graduate students) suggest that the system is an effective way to adjust notification state. Users need not make explicit gestures or utterances to change notification state; they need only specify the preferred behavior of the environment. (A forthcoming memo will describe studies currently underway.)

```
1. if person ?p in zone stool-1
   then person ?p in zone table

2. if person ?p in zone stool-2
   then person ?p in zone table

3. if number of people in zone ?z > 1
   then meeting in zone ?z

4. if person ?p in zone ?z
       and meeting in zone ?z
   then person ?p in meeting

5. if person ?p in meeting
       and message-delivery-event for ?p ?msg
   then deliver-without-display ?p ?msg

6. if deliver-without-display ?p ?msg
       and person ?p in zone ?z
   then notify agent for default-display-device for ?p in ?z
             "deliver-without-display" ?msg
```

Fig. 14. Example context inference and message delivery rules (in pseudo-code); rules are implemented in Joshua [31]; ?x indicates a variable

The above description illustrates two important points: (1) activity zones can be used to deliver context-dependent information, and (2) the blackboard and context inference system enable user-specified aggregations of zones and delivery preferences stated in terms of high level context descriptions.

7 Discussion and Future Work

The concept of activity zones – regions of location context formed by observing user behavior – is a broad one, and we have only begun to explore its full extent. In addition to ongoing user studies of the current prototype, we are continuing to explore issues of user interaction focus on such questions as: which device and application behaviors are useful, how do users specify those behaviors, how zones are displayed to users. We also are exploring the issues of how our activity zone system selects default preferences for anonymous people, or selects preferences in a space inhabited by multiple people, each of whom may have their own preferences.

We anticipate being able to use information about furniture identity and locations to augment our context inference system with simple activity models representing such information as "meetings often happen at tables", "informal meetings often happen in comfortable chairs". We could use the perceptual features from our tracking system, plus identity of furniture objects, to index into a catalog of higher level contexts representing such activities as being in a meeting or reading (e.g., as in [32]). With extra information about objects and activities, we would be able to infer users' activities more accurately, thus increasing the relevance of task-related information and services provided by the workspace.

In building a catalog of higher level contexts, we plan to augment our observations of people in everyday work situations by investigating activity theory [5, 16], research on understanding how people work [14, 29], and how workspace design affects people's work [14, 20, 34]. This body of literature, along with our observations, will provide us with examples of social interactions that help define context for people.

An activity zone map may not be relevant if a physical space is rearranged, as shown, for example, in Figs. 9 and 10. In such circumstances, the system must adapt the activity zone map. One could consider two approaches: either save an image of a space for each of several activity zone maps, and periodically check that the image matches the current furniture configuration; or record furniture type and location in or near particular zones, then use a furniture tracker to notice a mismatch between original furniture locations and new locations. Since furniture tracking is a challenging computer vision problem, the first approach may be easier to implement. With easily distinguishable furniture, the second approach may be advantageous since it allows for more flexibility in space layout. We plan to explore both of these approaches.

We are keenly aware that many people are uncomfortable having their activities observed by cameras, and privacy issues deserve attention when designing context-aware systems, e.g., [1]. In our work to date we have found that it is important to have computationally transparent perceptual systems so that users are aware of the kind of image information that is processed and stored. Abstract trajectory data with voluntary and explicit identity assertion is generally considered less invasive of privacy than full motion video. We plan to investigate methods for real-time visualization of tracker state.

We have shown that our concept of activity zones can provide valuable context cues for context-aware applications. So far we have explored this idea with relatively simple applications – device control and message delivery. We plan to continue our investigations by using activity zones with other applications, for example, retrieving information based on a predicted context. A zone's information about motion could be used to predict where people are likely to be sitting or standing. This predictive capability may prove valuable with tasks that are more structured than those usually found in an office environment. In a laboratory setting, for example, scientists often engage in an ordered set of activities, centered around various pieces of laboratory equipment. Context-aware computing researchers are building applications to support such laboratory activities, e.g., [4]. The activity zones in such a setting would center on laboratory equipment stations, and could be mapped to an ordered list of activities in order to predict likely transitions between zones. By anticipating what scientists may do next, a support system could initiate start up procedures when necessary or gather additional information needed at a particular station.

Similarly, we believe that the activity zone concept has utility in the structured domain of theatre performance. We are in the beginning stages of a collaboration with the Royal Shakespeare Company, and plan to explore the use of the activity zone concept in tracking actors, identifying particular scenes in the progression of a play, and controlling lighting [28]. We also are talking with physicians at a local hospital about the possibility of using the activity zone concept to track the progression of activities in an operating room.

In these and other applications, we believe that activity-based location regions provide a richer level of context information than has been previously available in location-based context-aware computing.

Acknowledgments

We thank members of the AIRE (Agent-Based Intelligent Reactive Environments) and Visual Interface groups for help in setting up the workspace and for their insightful conversations about this research. We also thank Randall Davis and Robert Laddaga for valuable comments on drafts of this paper. This research is sponsored by MIT Project Oxygen. We are grateful for support from the MIT Oxygen Alliance Partners: Acer, Delta, Hewlett-Packard, Nokia, NTT, and Philips.

References

1. Ackerman, M., Darrell, T., and Weitzner, D.J.: Privacy in Context. Human-Computer Interaction 16 (2001) 167-176.
2. Addlesee, M., Curwen, R., Hodges, S., Newman, J., Steggles, P., Ward, A., and Hopper, A.: Implementing a Sentient Computing System. IEEE Computer 34 (2001) 50-56.
3. Alexander, C., Ishikawa, S., and Silverstein, M.: A Pattern Language: Towns, Buildings, and Construction. New York: Oxford University Press (1977).
4. Arnstein, L.F., Sigurdsson, S., and Franza, R.: Ubiquitous computing in the biology laboratory. Journal of the Association for Laboratory Automation (JALA) 6 (2001).
5. Bannon, L.J. and Bødker, S.: Beyond the Interface: Encountering Artifacts in Use. In: Designing Interaction: Psychology at the Human-Computer Interface. J.M. Carroll (ed.). Cambridge University Press, New York (1991) 227-253.
6. Beigl, M. (ed. Special Issue on Location Modeling in Ubiquitous Computing. Personal Ubiquitous Computing. Vol. 6 (2002).
7. Brumitt, B., Krumm, J., Meyers, B.B., and Shafer, S.: Ubiquitous Computing and the Role of Geometry. IEEE Personal Communications 7 (2000) 41–43.
8. Brumitt, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S.: EasyLiving: Technologies for Intelligent Environments. In: Proceedings of the 2nd International Conference on Handheld and Ubiquitous Computing: Springer-Verlag (2000).
9. Coen, M., Phillips, B., Warshawsky, N., Weisman, L., Peters, S., and Finin, P.: Meeting the Computational Needs of Intelligent Environments: The Metaglue Environment. In: Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments (MANSE 99). Dublin, Ireland: Springer-Verlag (December 1999).
10. Crabtree, A. and Rodden, T.: Routine Activities and Design for the Domestic. University of Nottingham Equator-02-037 (2002).

11. Demirdjian, D., Tollmar, K., Koile, K., Checka, N., and Darrell, T.: Activity Maps for Location-Aware Computing. In: Proceedings of IEEE Workshop on Applications of Computer Vision (WACD2002) (December 2002).
12. Dey, A.K. and Abowd, G.D.: Towards a Better Understanding of Context and Context-Awareness. In: Proceedings of the What, Who, Where, When, and How of Context-Awareness Workshop,CHI 2000: ACM (2000).
13. Dey, A.K., Abowd, G.D., and Salber, D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human-Computer Interaction 16 (2001) 97-166.
14. Duffy, F. and Powell, K.: The New Office. London: Conran Octopus Ltd (1997).
15. Engelmore, R. and Morgan, T. (eds): Blackboard Systems. Addison-Wesley, Reading, M A (1988).
16. Engestrom, Y., Miettinen, R., and Punamaki, R.-L. (eds): Perspectives on Activity Theory. Cambridge University Press, New York (1998).
17. Hall, E.T.: The Hidden Dimension. New York: Doubleday (1969).
18. Hanssens, N., Kulkarni, A., Tuchida, R., and Horton, T.: Building Agent-Based Intelligent Workspaces. In: Proceedings of The International Workshop on Agents for Business Automation. Las Vegas, NV (June 2002).
19. Harle, R.K. and Hopper, A.: Using Personnel Movements for Indoor Autonomous Environment Discovery. In: First IEEE International Conference on Pervasive Computing and Communications (PERCOM 2003). Fort Worth, TX: IEEE (2003).
20. Hartkopf, V., Loftness, V., Aziz, A., Shankavaram, J., and Lee, S.: The GSA Adaptable Workplace Laboratory. Lecture Notes in Computer Science 1670 (1999) 29-44.
21. Hertzberger, H.: Lessons for Students of Architecture. Rotterdam: Uitgeverij Publishers (1991).
22. Hightower, J. and Borriello, G.: A Survey and Taxonomy of Location Systems for Ubiquitous Computing. University of Washington, Computer Science and Engineering: Seattle, WA (2001).
23. Hightower, J., Brumitt, B., and Borriello, G.: The Location Stack: A Layered Model for Location in Ubiquitous Computing. In: Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002). Callicon, NY (2002).
24. Hildebrand, G.: The Wright Space: Pattern and Meaning in Frank Lloyd Wright's Houses. Seattle: University of Washington Press (1991).
25. Hillier, B. and Hanson, J.: The Social Logic of Space. New York: Cambridge University Press (1984).
26. Johanson, B., Fox, A., Hanrahan, P., and Winograd, T.: The Event Heap: A Coordination Infrastructure for Interactive Workspaces. In: Fourth IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002). Callicoon, NY (2002).
27. Junestrand, S., Keijer, U., and Tollmar, K.: Design Patterns for Private and Public Digital Spaces in the Home Environment. International Journal of Human-Computer Studies 54 (2001) 753-778.
28. Kennan, F.: Lights, Hologram, Action! Business Week, May 26 (2003) 97.
29. Kirsh, D.: The Context of Work. Human-Computer Interaction 16 (2001) 305-322.
30. Lahlou, S.: Observing Cognitive Work in Offices. In: Cooperative Buildings: Integrating Information, Organizations, and Architecture. N. Streitz, et al. Editors. Springer, Heidelberg (1999) 150-163.
31. Rowley, S., Shrobe, H.E., Cassels, R., and Hamscher, W.: Joshua: Uniform Access to Heterogeneous Knowledge Structures, or why Joshing Is Better than Conniving or Planning. In: AAAI (1987).
32. Turner, R.M.: Context-Mediated Behavior for Intelligent Agents. Int. J. Human-Computer Studies 48 (1998) 307-330.
33. Winograd, T.: Architectures for Context. Human-Computer Interaction 16 (2001) 401-419.
34. Zelinsky, M.: New Workplaces for New Workstyles. New York: McGraw-Hill (1998).

Context-Aware User Authentication – Supporting Proximity-Based Login in Pervasive Computing

Jakob E. Bardram, Rasmus E. Kjær, and Michael Ø. Pedersen

Centre for Pervasive Computing
Department of Computer Science, University of Aarhus
Aabogade 34, 8200 Århus N, Denmark
{bardram,rkjaer,michael}@daimi.au.dk

Abstract. This paper explores computer security in pervasive computing with focus on user authentication. We present the concept of *Proximity-Based User Authentication*, as a usability-wise ideal for UbiComp systems. We present a context-aware user authentication protocol, which (1) uses a JavaCard for identification and cryptographic calculations, (2) uses a context-awareness system for verifying the user's location, and (3) implements a security fall-back strategy. We analyze the security of this protocol and discuss the tradeoff between usability and security. We also present our current implementation of the protocol and discuss future work.

1 Introduction

The notion of ubiquitous and pervasive computing implies a shift in the use of computers. We are going from the personal computing paradigm, where each user is using a personal computer, to the pervasive computing paradigm, where computers are available in huge numbers, embedded in everyday artifacts, like phones, furniture, cars, and buildings. Hence, each user is using many personal computing devices, and at the same time, the same publicly available device is used by many users. This shift from a 'one-to-one' to a 'many-to-many' relationship between users and computers sets up some new usability challenges for computer security, especially user authentication. Contemporary user authentication schemes involve typing in usernames and passwords. When using a personal computer, typing in username and password is straightforward, but still it poses substantial usability problems in some work environments, like hospitals [4, 23]. In the pervasive computing paradigm, these usability problems are increasing, because the user is using many computers. Imagine that a user would need to type in username and password on all 'pervasively' available computers before he could start using them. Clearly, if the pattern of login and logout is not considered a usability problem today, it will most certainly become one in the years to come.

In this paper we describe our response to such usability problems of user authentication in a pervasive computing environment. Our aim is to support what we have termed *proximity-based login* (see also [11]), which allows users to be authenticated on a device simply by approaching it physically. The idea of enabling users to access a computer by simply walking up to it has a long history in ubiquitous computing research. This idea of proximity-based login can be traced back to the pioneering work on

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 107–123, 2003.

the Active Badge System, which could be used to 'teleport' an X Window session to a display located in front of the user [27, 5]. Similarly, the AT&T Active Bat system [28] was used to create 'Follow-me Applications' where the user-interfaces of the user's application can follow the user as s/he moves around. The application is shown on the display that is deemed to be in front of the user as established via the user's Active Bat [16]. The idea has later been adopted by the Microsoft 'EasyLiving' project [7]. In other pervasive computing environments different types of physical tokens are used as user identification. In the BlueBoard project at IBM, a HID brand reader is used [22, 21]. In the AwareHome at Georgia Tech RFID tags provide identity of individuals near commonly used monitors [19], and at FX PAL the Personal Interaction Points (PIPs) System uses RFID cards that stores the users identification as well as passwords (the latter in encrypted form) [26]. Ensure Technologies' XyLoc system [15] uses short-range radio communication between a personal token and a PC to establish the proximity of a user and to unlock the PC by pressing a button on the token, not unlike the ones used to remotely unlock cars.

Common to these systems is the lack of proper security mechanisms that can effectively ensure a secure user authentication. In case of theft of a token, or by recording and replaying the communication between the token and the reader, an adversary can access the system and impersonate the legitimate user. The Smart Badge platform [24] uses some improvements to reduce the problem of stolen tokens. The method is to use a badge that can detect when it is no longer being carried. The link between the badge and a particular user can then be removed, and the badge can no longer be used for authentication purposes. It is however difficult to judge from the paper exactly how such a smart badge can sense whether is is being carried by its legitimate user. The Zero-Interaction Authentication method of [12] takes an approach similar to the XyLoc system. A token is used to gain access to a laptop with an encrypted file-system. To verify the user, s/he is required to enter a PIN code on the token before s/he can start using it.

There is often an inherent tradeoff between usability and security. User authentication mechanisms tend to be either secure, but less usable, or very usable, but less secure. It is our aim to try and combine the two standpoints and suggest a *context-aware user authentication mechanism* that is very usable as well as sufficiently secure for use in settings, where security matters – like a hospital environment. Traditionally a user authentication mechanism is considered secure if it is a combination of something the user *has* (e.g. a smartcard), something the user *knows* (e.g. a password), or something the user *is* (i.e. a fysiological trait) [25]. Our design of a user authentication mechanism is based on supplementing well-known user authentication mechanisms with knowledge about the context and location of the user. In line with Denning [14] we thus suggest location-based authentication and introduce *'location'* as a fourth element in a user authentication mechanism. The paper starts by outlining the background and motivation for the design of context-aware user authentication. Section 3 presents our design of a proximity-based user authentication mechanism and its degree of security is analyzed in section 4. Section 5 describes our current implementation of the proposed protocol, and section 6 discusses our work before the paper is concluded.

2 Background and Research Methods

The work on proximity-based user authentication takes place within the research area of *Pervasive Healthcare* [10]. A central area of research is to design and develop pervasive computing technologies for the use at hospitals. We approach this challenge in two ways. On the one hand we conduct ethnographic studies of clinical work in hospitals and how technology is used here. And on the other hand we engage clinicians in experimental design and development of technologies in various types of workshops in our laboratory.

2.1 Troubles with Login

In a study of the use of Electronic Patient Records (EPR) at a large metropolitan hospital, we observed a number of usability problems associated with user authentication [4]. The EPR was accessed through PCs distributed within the hospital, and it had a traditional login system with usernames and passwords. Thus, whenever a clinician should access patient information s/he had to log in and out on different PCs. Due to the way the PCs were deployed and the nature of the work in hospitals, it was not uncommon that a nurse, for example, would log in 30 times a day. Because this was a highly cumbersome thing to do in a hectic environment, workarounds were established. For example, users would avoid logging out, enabling them to return to the PC without logging in later; passwords were shared among users and made very easy to remember ('1234' was the most used password at the hospital); and users would often hand over user sessions to one another, without proper logout and login. Hence, what was designed to be a secure system (with traditional username and password user authentication) was suddenly turned into a highly insecure system, because of obvious usability problems.

2.2 Activity-Based Computing

Even though this EPR system in no way can be termed as 'pervasive technology', our study of its use has highlighted how essential user authentication is to the design of pervasive computer support for medical work in hospitals. In our second line of research we actively design and develop pervasive computing technologies for hospital work. A central component in this effort is a basic runtime infrastructure, which supports Activity-Based Computing (ABC) [11]. The basic idea of activity-based computing is to represent a user's (work) activity as a heterogeneous collection of computational services, and make such activities available on various stationary and mobile computing equipment in a hospital. Clinicians can initiate a set of activities, and access these on various devices in the hospital. For example, a nurse can use the computer in the medicine room to get some medicine, and later when giving this medicine to the patient she can restore the patient and medicine data on the display in the hospital bed. We have built prototypes of wall-size displays, displays embedded in tables, built-in computers in hospital beds, and we are using various mobile equipment like TabletPCs and PDAs. Figure 1 illustrates how a physician is using a wall-based display in a conference situation. Thus, activity-based computing allows users to carry with them, and restore, their work on heterogeneous devices in a pervasive computing environment. Central to this is clearly that users need to be authenticated on every device they want to use, and easy login is hence a core challenge in the concept of activity-based computing.

Fig. 1. A physician is using a wall-based display in a conference situation. In her hand she is holding a *Personal Pen*, which is used to authenticate her to the computer. An active badge woven into her white coat (not visible) is revealing her location to a context-awareness system.

Our design is based on participatory design sessions and workshops with a wide range of clinicians, including physicians, radiologists, surgeons, and different types of specialized nurses. All in all 12 such workshops were conducted, each lasting 4-6 hours having 6-10 participants each. Various aspects of designing support for clinical work were discussed, including the login mechanisms. Several user authentication mechanisms were designed, implemented, and evaluated in these workshops.

2.3 Requirements for a Pervasive Computing User Authentication Mechanism

Based on existing research within UbiComp, our studies of medical work, and our experimental design effort with end-users, we can list the following requirements for a user authentication mechanism in a pervasive computing environment.

- *Proximity-based* – Work at a hospital is characterized by busy people who are constantly moving around, and are engaged in numerous activities in parallel. Easy and fast login was thus deemed a fundamental prerequisite for the success of a distributed, pervasive computing infrastructure, embedded in walls, floors, tables, beds, etc. The usability goal in our workshops reached a point where the user should do nothing to log in – s/he should simply just use the computer, and the computer would know who the user was.

- *Secure* – Clinical computer systems store and handle sensitive, personal health data for many patients. It is therefore of utmost importance that these systems are protected from unauthorized access. Hence, pervasive computer systems in a healthcare environment require secure user authentication. This is used to set up the right user authorizations for reading and altering clinical data. For example, clinicians may only access clinical data related to patients, they are treating, and may not, for example, access clinical data on arbitrary persons. This is done to ensure the privacy of patients. Similarly, only physicians may prescribe medicine. One of our early designs for a user authentication mechanism was based on RFID tokens alone. This mechanism, however, was abandoned for obvious security reasons.
- *Active gesture* – We also experimented with a login mechanism that automatically would transfer a user's on-going session to a display nearby him – much like the 'Follow-me' application using the Bat system [16]. This, however, turned out to be a less useful design. The problem was that often a clinician would enter a room, where numerous computing and display devices would be available. For example in a radiology conference room, there would be several wall-based displays, a wide range of desktop computers, and an interactive table where images can be displayed and manipulated. It was unclear from monitoring the location of the user, which of such displays he would like to use – or whether he wanted to use a computer at all. Therefore the authorization mechanism must be based on an active gesture nearby the display or devices that the user wants to use.
- *Support for logout* – During our experiments we discovered that the process of logging out a user is equally important. Clinicians would often have to hurry on, and would simply walk (or run) away from an ongoing session. In this case, automatic logout was deemed important. Even though this is normally not considered to be part of a user authentication mechanism, we argue that logout has to be considered as a part of the design as well.

3 Context-Aware User Authentication

There are three key principles in our design of a context-aware user authentication mechanism. First, it uses a physical token used for active gesturing and as the cryptographic basis for authentication. Second, it uses a context-awareness system to verify the location of the user, and to log out the user when s/he leaves the computer(s) in a certain place. Third, it contains 'fall-back' mechanisms, so that if either of the two components in the system falls out, the user authentication mechanism switches to other mechanisms. If the context-awareness infrastructure is unreachable for any reason, the user is requested to enter his password when trying to log in. If the token cannot be accessed for any reason, the user is requested to enter both his username and password, as usual. Hence, in case of system failure, security is not compromised, and the system is still usable. We shall return to a more detailed security analysis below.

Our current design uses smart card technology and in the rest of the paper we will present this design based on a smart card as the physical token. Furthermore, we shall only consider the part of the authentication protocol that involves the smart card. Standard authentication using username and password is not further discussed. The two basic components in the context-aware user authentication mechanism are (1) a secure user authentication protocol between a computer and a smart card, and (2) a distributed computing context-awareness infrastructure.

3.1 Authentication Protocol

The authentication protocol is running on a JavaCard [3]. Each client is equipped with a card reader and the protocol is executed every time a user inserts his card into the reader on the client. In order to use the card for authentication, the following information is stored on the card:

– An id for the user the card belongs to.
– The user's $password$.
– The user's pair of a secret key (K_S) and public key (K_P).

When the card is issued, an applet[1] for authentication is stored on the card, and initialized with the user's $password$ and the id of the user. When the applet is initialized, the card creates an RSA key-pair. The secret key (K_S) is stored on the card and the public key (K_P) is stored in a central server along with the id of the user. The authentication protocol is illustrated in figure 2 and consists of the following steps:

1. The client receives notification that user P is in the room (optional).
2. The user places his smart card in the card reader.
3. The client requests the id from the smart card.
4. The client looks up the person in the Context Server based on the id from the card.
5. There are two distinct cases based on the probability that the user is in the same place as the client.
   Case A: The probability is greater that a certain threshold.
   – The smart card is asked to verify that it holds the user's secret key, K_S.
   Case B: The location of the user is not sufficiently sure.
   – The computer asks the user to enter his $password$.
   – The smart card accepts or rejects the user based on the password.
6. The user is either denied or allowed access.

In case A, where the user is known to be in the room, the client verifies that the smart card knows the user's secret (private) key K_S by generating a random 20 byte "nonce", N, and sends it to the card. The card then sends back the signature under the private key, $sig\,(K_S, N)$, of N, and the client uses the corresponding public key, K_P, to verify that the signature is correct.

In case B, the user is not known to be in the room. The client asks for the user's password, concatenates it with a 20 byte nonce N, encrypts it under the user's public key, $E\,(K_P, password + N)$, and sends it to the card. Since the card knows the secret key, it can decrypt this message. It then compares the received password with the one stored on the card. If they match the card returns a byte R and a signature on R concatenated with N, $R + sig\,(K_S, R + N)$, where R equal to 1 is accept and 0 reject. The client uses the public key, K_P, to verify the signature.

[1] Programs running on JavaCards are called applets. They bear no resemblance with the applets running in web browsers, except for the name.

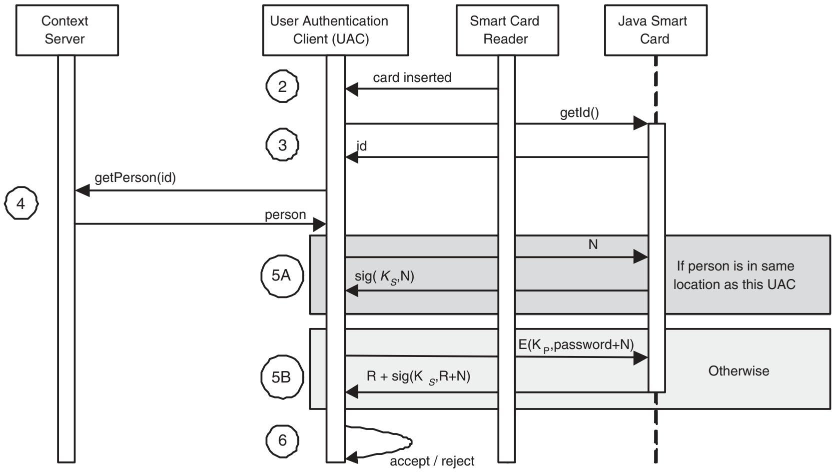


Fig. 2. Interaction Diagrams for the Authentication Protocol. Case A – The person is in the same location as the User Authentication Client (UAC). Case B – The person is not in the same place, and the user is requested to enter his or her password.

3.2 Infrastructure

The system architecture for the context-awareness infrastructure is illustrated in figure 3, and consists of the following main components:

- *Context Monitors* – A range of hardware and/or context data specific processes, which register changes in the environment. The monitor adapts this context information according to the data model in the Context Server. Examples of context monitors are location monitors based on monitoring RFID tags attached to items in the environment, or WLAN monitors that try to locate WLAN-based equiptment. Other monitors might gather information about temperature, planned activities in users' personal calendars, or try to identify people in a room based on their voices. Currently we mainly use a Portal RFID antenna (see figure 5) for locating people. When a portal monitor scans an RFID tag, the RFID adapter translates the 64 bit tag id into telling the context server that a specific entity has been seen at a specific location. Even though our current location mechanism (also) is based on something the user has (a RFID tag), the architecture of our context-awareness infrastructure enables us to create other location monitors based on e.g. audio or video.
- *Context Server* – The Context Server contains a simple data structure that stores information about 'Entities' in the environment. Entities are basically people, places, or things, but this structure is extensible and all kinds of context data can be stored by implementing some simple interfaces.

From a client's point of view there are basically two ways to connect to the Context Server. On the one hand, a client can look up an entity and ask for its context information, like location. For example, a client can request a person and ask for the location.

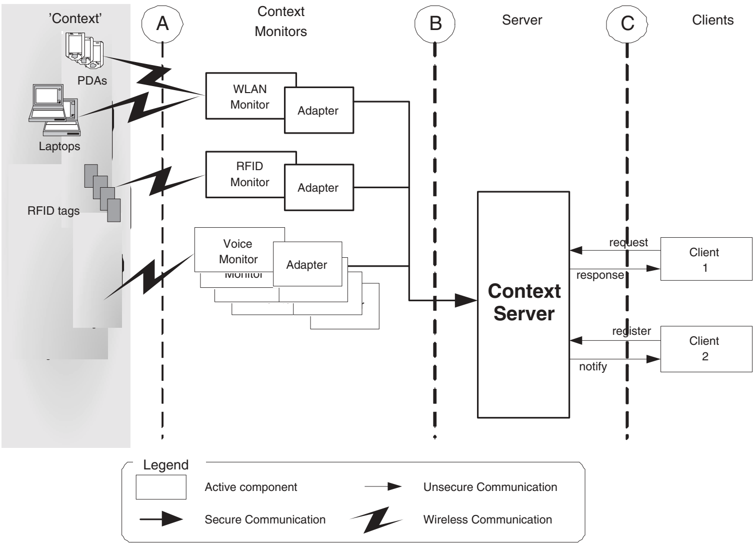


Fig. 3. System architecture for the Context-Awareness Infrastructure.

On the other hand a client can register itself as a listener to an entity, and it will hereafter receive notifications when changes to this entity take place. For example, a client running on a stationary computer (e.g. a wall-sized display) can register itself as listener for changes to the place in which it is located. Whenever entities enter or leave this place, the client will be notified.

In our authentication protocol, the context server is asked for a person's location. The confidence in the answer from the server can be divided into addressing two questions:

1. *How accurate is the location data?* – Whenever the context server provides a location of an entity, it estimates the accuracy of this location. Thus, the context server embeds an *accuracy algorithm*. In step 5 of the protocol, the accuracy of the location estimate is compared against a configurable threshold value.
2. *Do we trust the location data?* – This is a question of whether we trust the information that is stored in the context server. Because monitors are the only way location data can be altered in the server, this reduces to a question of trusting the Context Monitors. To prevent non-authorized monitors to access and update the context server we require the monitors to authenticate themselves to the server. Hence, our context-awareness architecture supports secure authentication and access in layer B in figure 3 between the monitors and the server. However, there is currently no security in layer A between the tokens and the monitors, because RFID tags do not have processing power to support a cryptographic setup. We shall return to the security consequences of this below.

The communication layer C between the Context Server and the clients is not secure. Hence, a non-authorized client can get access to the server and read data from it.

This might be a problem from a privacy point of view, but from a security point of view clients cannot alter location data in the context server. We shall return to the security analysis below.

4 Security Analysis

We will now take a look at the security of this authentication mechanism. The goal of a person trying to break the system is to authenticate as a legitimate user. This person is called the adversary. We will make the assumption that the adversary has access to all information stored about the user, except from access to information stored on the smart card (the private key and the user's password). We will also assume that communication between the Context Monitors and the Context Server is secure and that only legitimate Context Monitors can access the Context Server. Finally we assume that the information stored on a smart card cannot be read or changed except through the designed interfaces[2].

4.1 Passive Attacks

In the passive attack scenario we assume that the adversary can monitor all communication between the smart card and the terminal. In the case where location data is based on a token the user has, the adversary can also monitor all communication between that location token and the Context Monitor. Using the information he acquires during this phase he will now try to impersonate the legitimate user after having acquired the user's smart card, both the smart card and the location token (if one is used) or none of them. We have identified the following passive attacks:

1. If the adversary acquires the smart card and is able to fake the location of the legitimate user (by stealing the location token or cheating other devices used to establish the location of the user) he can authenticate as the legitimate user since to all parts of the system he is that user.
2. If he acquires only the smart card he can authenticate as the legitimate user only in the location where the user is actually present.
3. If the user is not present he cannot authenticate as that user even though he has the smart card, unless he also knows the user's password. This is because the protocol used is a secure one-way authentication protocol. Proof of this can be found in appendix A.
4. If he does not have the smart card, it is impossible for him to do anything, since the terminal will ignore him when he approaches it.

4.2 Active Attacks

Where the passive adversary could only look at messages between the location token and the Context Monitor and the smart card and the terminal, the active adversary can

[2] It is up to the manufacturer to protect against physical attacks on the card [2, 18].

drop, change, inject or completely replace any of these messages. He can also create his own smart card using the information he obtains. The goal is the same, though: After having done this for as long as he wants he will now try to impersonate the legitimate user. We have identified the following active attacks:

1. The adversary can retransmit the "I'm here" message from a location token to a Context Monitor or he can trick some other part of the location system to make the Context Server believe that the legitimate user is present. If the adversary has the smart card he can now authenticate as the legitimate user.
2. Replaying a value created by the legitimate user in another run of the protocol between the smart card and the terminal does not work since the protocol is a secure one-way authentication protocol and hence not vulnerable to replay attacks. More specifically it comes from the fact that the probability that the terminal will use the same value of N in another run of the protocol is negligible. This is true for both case A and case B.
3. An idea could be to trick the smart card into decrypting the value containing the password. Since signing using RSA is encrypting with the private key we could send $E(K_P, N+password)$ from case B to the legitimate user's smart card instead of N in case A. The smart card will then sign this value and return the signature, which will actually be the decryption of the password. This attack does not work since the smart card will never decrypt and return anything it receives. This means that the smart card cannot be used as a *Decryption Oracle*. In this case the smart card signs a hash of that value instead of the value itself. Since cryptographic hash functions are believed to be one way, the adversary cannot control what value is signed.
4. Creating a new smart card with a known password will not work since the public key will not be known to the server. If we use a legitimate user's public key we would not have the private key and could not make a smart card that can decrypt the first message in case B or make a signature that can be verified using a legitimate user's public key, which is required in both case A and B.

Another attack against this protocol is a *proxy attack* where the adversary tricks the user into using a fake terminal. This allows the adversary to read his password or to perform a man-in-the-middle attack where he can authenticate as the legitimate user without having the user's smart card. This requires that both the user and the adversary is running the same version of the protocol.

If they are both running the protocol from case A, the adversary simply forwards N to the legitimate user's card, which will return a signature on that value, and the adversary can now send this to the real terminal. The adversary is hereby granted access in the name of the legitimate user. Figure 4 illustrates this attack.

If we are in case B, this is a bit trickier since the adversary must know the correct password in order to succeed. This is because the nonce the real terminal sends is combined with the password and encrypted. In order for this attack to work, the smart card has to sign the nonce the real terminal sent, which means that it must also accept the password entered on the real terminal by the adversary. However this is not a problem for the adversary since the fake terminal can also read the user's password. This attack is also illustrated in Figure 4.

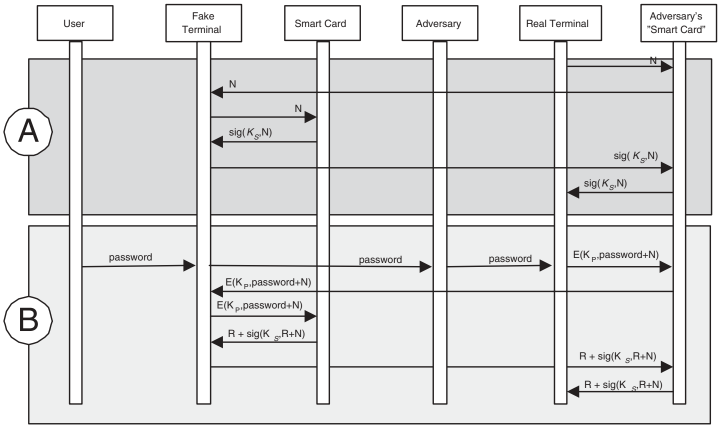


Fig. 4. Proxy attack where an adversary has put in a fake smart card terminal to be used by an ignorant user.

4.3 Summary

It is possible for the adversary to authenticate as a legitimate user by doing one of the following:

1. Steal the smart card and fake the location of a legitimate user by
   (a) Replay the “I’m here” message from a location token
   (b) Trick other parts of the location system in various ways
2. Steal the smart card and be in the same room as the legitimate user.
3. Steal the smart card and acquire the user’s password somehow.
4. Perform one of the two proxy attacks mentioned above.

We realize that the real weakness in this protocol is the location data. If that can be faked, all you need is to acquire the smart card of a legitimate user, but without knowing the details on how location data is obtained you can not do a thorough security analysis of that problem. If location data is only based on some token the user has, it can be stolen, if such a location token does not use some kind of cryptography it is vulnerable to a reply attack, if you use voice recognition it might be fooled by a recording of the user’s voice, etc. The proxy attacks are bad as well, but require more resources and knowledge to succeed and there are ways to prevent them (see section 6).

5 Current Implementation

Our current implementation consists of five parts:

- An installer.
- The authentication applet.
- A rough prototype of a Personal Pen
- The Context Server and Monitors
- A client that runs the authentication protocol.

The *Installer* installs the applet on the card with the help of IBM JCOP tools [17]. Then it retrieves the public key from the card, and stores it as a key-value pair on the person object in the Context Server. In our current implementation, the authentication applet uses 512 bit RSA keys. This can easily be changed since the cards also support 1024 and 2048 bit keys. Encryption is done in PKCS#1 mode and signatures are made by taking a SHA1 hash of the data to be signed and then this hash is encrypted with the private key in PKCS#1 mode. We run the applet on the dual-interface OpenPlatform compliant JavaCard designed by IBM, which supports both contact-based and contactless connections to the card reader. We use Philips Semiconductors' MIFARE PRO contactless reader [20] (see figure 5) as well as standard OpenPlatform Smart Card readers in e.g. keyboards.

Our current design of the *Personal Pen* is just to glue a JavaCard to a Mimio Pen, which is used at the wall display in figure 1. This form factor is not particularly appealing (see figure 5), but it is hard to change the form and size of the card, because the antenna is embedded in the plastic that surrounds the chip itself. However, the size of the chip in these cards does not prevent it from being embedded in a pen at the factory. Our ideal hardware design would be a smart card chip embedded in a pen, and the reader embedded in the touch-sensitive layer on a display. Putting the tip of the pen at the display would then correspond to inserting a card in a card reader. In this way we would be able to authenticate the user every time s/he is doing anything on the display. The *User Authentication Client* runs as a part of the Activity-Based Computing infrastructure, and it simply waits for a card to be inserted in the reader, and then runs the protocol.

Fig. 5. *Left* – The ScanGate RFID Long Range Portal Antennas from Datatronic. *Right* – The dual-interface OpenPlatform JCOP JavaCard designed by IBM using the Philips Semiconductors' MIFARE PRO contactless reader. The JavaCard is glued to a Mimio pen, which is used at the wall-display in figure 1.

As for the *Context Server*, we have currently implemented the two monitors shown in figure 3. The WLAN monitor monitors the WLAN base station infrastructure in our lab and can tell the cell-based location of IEEE802.11b networked devices. Various types of RFID monitors can monitor passive RFID tags in the environment. We currently use the Portal antennas shown in figure 5 to determine the location of persons equipped with RFID tags. We use the Long Range RFID Antennas from Datatronics [13].

The current implementation of the *Accuracy Algorithm* is very simple. It reduces the accuracy of the location estimate by 1% every minute. Thus, if a person has passed a portal 10 minutes ago, s/he is in this location with a probability of 90%. The User Authentication Client is also considered a trusted monitor (we call it a 'Login Monitor') and can hence reveal the user's location every time s/he logs in. The secure authentication of monitors has not been implemented using proper cryptograpic protocols. However, since monitors run on standard PCs there are already well-known ways of doing this using e.g. a PKI setup over secure IP. Hence, this was not necessary in order to make a proof-of-concept.

6 Discussions and Future Work

By combining a context-awareness sub-system with a personal smart card, we have designed and implemented a proof-of-concept of a proximity-based user authentication mechanism, which is both user-friendly and secure (c.f. section 2.3). One could argue that it seems like an unnecessary effort to implement a context-awareness system in an organization just to help users log in. However, on the one hand we argue that providing easy login is essential to maintaining a smooth flow of work in a hospital (see also [4]). And on the other hand, we envision that a context-awareness sub-system is already in place in a hospital for many other reasons, and we have demonstrated how such a system can help realize the vision of 'proximity-based user authentication'.

From a security point of view, the authentication mechanism presented here, allows for some additional attacks as compared to the traditional use of smart cards in conjunction with passwords (attack no. 3 is equally relevant in both cases). We consider attack no. 2 highly unlikely, which leaves us with attack no. 1 and 4. Our current implementation cannot cope with attack no. 1.a, except for simple revocation of smart card and location token (the RFID tag) when the theft is discovered. However, determining a person's location could be based on something that is not token-based, for example video or audio. We currently plan to develop a voice-print monitor that can identify and locate a user based on voice. Such a voice monitor is not subject to theft, and methods for avoiding playback exist. In attack no. 1.a the adversary tells a Context Monitor that the user is present by actively replaying the user's ID. While we cannot completely prevent this given the computational power available in RFID tags, we can reduce the possibility of this attack succeeding by doing some additional checks in the Context Server. For example, the user cannot be present on two different locations at the same time, move from one location to another in less than a specified amount of time, or be present in the hospital outside his/her normal working hours. If any of these events occur, uncertainty in location estimation is increased and the authentication protocol will ‘fall back’ to password authentication. Attack no. 4 could be considered the most dangerous attack since it does not require access to the smart card or the location token. The attack against case A is easily prevented by disallowing the user to be present in two different locations simultaneously. Against case B the attack is more difficult to prevent. One solution could be to use biometrics instead of the password. That way the adversary cannot enter the password on the real terminal. This would also reduce the risk of the adversary aquiring the user’s password or even if he could aquire the binary data generated by the biometric equipment he cannot feed it to the real terminal easily. Another solution to the proxy attack could be to use some kind of *Distance-Bounding Protocol* [9] to ensure that the user is close to the terminal he is trying to log on to.

A solution to the active replay attack (attack no. 1.a) could be to have location tokens to make a secure authentication when revealing itself to the context monitors (communication link A in figure 3). This could be done by using public key cryptography, almost similar to the smart card authentication. Another approach would be to use an authentication protocol that relies on secret key cryptography since it does not require the same amount of processing power. We are currently designing radio-based tokens that carry enough processing power and memory to make a secret key authentication to their context-awareness monitors.

7 Conclusion

In this paper we have pointed to the need for considering security, and especially user authentication, as a central aspect of UbiComp technologies and applications. It is of crucial importance that users can access pervasively available computers easily, if the vision of UbiComp shall become alive. In the paper we introduced the concept of *proximity-based user authentication* in a pervasive computing environment, where a user is authenticated to a device by just approaching it. Our aim was to create a user-friendly user authentication mechanism, which is sufficiently secure to be used in settings where security is important – like in a hospital. The suggested solution is based on using two different mechanisms for user identification and verification. A user is identified by presenting his Java Smart Card to the computer whereafter his correct presence is verified through a context-awareness system. If the context-awareness system is unavailable or cannot localize a user, the user is required to enter his password – this is identical to the standard use of smart cards.

Through our security analysis we have argued for the degree of security in our solution. This analysis showed that our solution is not completely secure. Especially, the solution is vulnerable to an active replay attack, where an adversary steals a user’s smart card and at the same time is able to replay a false “I’m here” message to a nearby Context Monitor. To accommodate this attack, the Context Server implements an accuracy algorithm that calculates the accuracy or probability of the estimated location of a person. In case this probability is below a certain threshold, the user is requested to enter a password. We argue that this authentication mechanism is sufficiently secure because it combines something the users *have* (the smart card) with the *location* of the user. This authentication protocol is hence and example of location-based autentication [14]. This is traditionally taken as sufficiently secure, if each of the two mechanisms is secure [25].

Acknowledgments

Many thanks to Michael Osborne at the IBM Zurich Research Laboratory, who has provided us with the JCOP JavaCard technology as well as with much practical help. Thanks to Kim Kortermand for help with the RFID Portal Antennas. This research has been partly funded by the Competence Centre ISIS Katrinebjerg.

References

1. G. D. Abowd, B. Brumitt, and S. Shafer, editors. *Proceedings of Ubicomp 2001: Ubiquitous Computing*, volume 2201 of *Lecture Notes in Computer Science*, Atlanta, Georgia, USA, Sept. 2001. Springer Verlag.
2. R. Anderson and M. Kuhn. Tamper resistance – a cautionary note. *The Second USENIX Workshop on Electronic Commerce Proceedings*, 1996.
3. M. Baentsch, P. Buhler, T. Eirich, F. Höring, and M. Oestreicher. Javacard – from hype to reality. *IEEE Concurrency*, pages 36–43, Oct.-Dec. 1999.
4. J. E. Bardram. The Trouble with Login – On usability and Computer Security in Pervasive Computing. Technical Report CfPC 2003–PB–50, Center for Pervasive Computing, Aarhus, Denmark, 2003. Available from http://www.pervasive.dk/publications.
5. F. Bennett, T. Richardson, and A. Harter. Teleporting – Making Applications Mobile. In *Proceedings of the IEEE Workshop on Mobile Computer Systems and Applications*, pages 82–84, Los Alamitos, CA, USA, 1994. IEEE CS Press.
6. G. Borriello and L. E. Holmquist, editors. *Proceedings of Ubicomp 2002: Ubiquitous Computing*, volume 2498 of *Lecture Notes in Computer Science*, Göteborg, Sweden, Sept. 2002. Springer Verlag.
7. B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer. EasyLiving: Technologies for Intelligent Environments. In P. Thomas and H. W. Gellersen, editors, *Proceedings of Handheld and Ubiquitous Computing, HUC 2000*, volume 1927 of *Lecture Notes in Computer Science*, pages 12–29, Bristol, UK, Sept. 2000. Springer Verlag.
8. M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems (TOCS)*, 8(1):18–36, 1990.
9. L. Bussard and Y. Roudier. Embedding distance-bounding protocols within intuitive interactions. 2003.
10. Center for Pervasive Healthcare. http://www.cfph.dk.
11. H. B. Christensen and J. Bardram. Supporting human activities – exploring activity-centered computing. In Borriello and Holmquist [6], pages 107–116.
12. M. D. Corner and B. D. Noble. Zero-interaction authentication. In *Proceedings of the eighth annual international conference on Mobile computing and networking*, pages 1–11. ACM Press, 2002.
13. Datatronic Long Range RFID Portal Antennas (297C-AT3M/S). http://www.datatronic-rfid.com.
14. D. E. Denning and P. D. MacDoran. Location-Based Authentication: Grounding Cyberspace for Better Security. *Computer Fraud and Security*, February 1996. Available from http://cosc.georgetown.edu/ denning/infosec/Grounding.txt.
15. Ensure Technologies. http://www.ensuretech.com.
16. A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The Anatomy of a Context-Aware Application. In *Proceedings of the 5th Annual ACM/IEEE Conference on Mobile Computing and Networking (MobiCom99)*, pages 59–68, Seattle, WA, USA, 1999. ACM Press.

17. IBM JCOP Smart Card Technology. http://www.zurich.ibm.com/csc/infosec/smartcard.html.
18. P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. *Lecture Notes in Computer Science*, 1666:388–397, 1999.
19. K. Nagel, C. D. Kidd, T. O'Connell, A. Dey, and G. D. Abowd. The Family Intercom: Developing a Context-Aware Audio Communication System. In Abowd et al. [1], pages 176–183.
20. Philips Semiconductors' MIFARE Demonstration System (MF EV500). http://www.semiconductors.philips.com/markets/identification/products/mifare/.
21. D. M. Russell, C. Drews, and A. Sue. Social Aspects of Using Large Public Interactive Displays for Collaboration. In Borriello and Holmquist [6], pages 229–236.
22. D. M. Russell and R. Gossweiler. On the Design of Personal & Communal Large Information Scale Appliances. In Abowd et al. [1], pages 354–361.
23. B. Schneider. *Secrets and Lies : Digital Security in a Networked World*. John Wiley & Sons, 1 edition, 2000.
24. M. T. Smith. Smart Cards: Integrating for Portable Complexity. *IEEE Computer*, pages 110–115, Aug. 1998.
25. A. S. Tanenbaum. *Modern Operating Systems*. Prentice Hall, 2 edition, 2001.
26. J. Trevor, D. M. Hilbert, and B. N. Schilit. Issues in Personalizing Shared Ubiquitous Devices. In Borriello and Holmquist [6], pages 56–72.
27. R. Want, A. Hopper, V. Falco, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, 1992.
28. A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communication Magazine*, 4(5):42–47, October 1997.

A Security Proof

In BAN logic [8] the authentication protocol looks like this:

Case A

1. $B \to A : N$
2. $A \to B : \{N\}_{K^{-1}}$

Case B

1. $B \to A : \{N, P\}_K$
2. $A \to B : \{N, R\}_{K^{-1}}$

Where A is the smart card, B is the terminal, N is a nonce, P is the password entered by the user, K is the public key of the smart card, K^{-1} is the private key of the smart card and R says whether or not P matched the password stored on the smart card. The proof will be split in two different cases.

We want to show that in both cases the terminal can be convinced that the legitimate user is present in the current run of the protocol. In the first case this is done by showing that the smart card knows the private key and the password that the user entered. In the second case this is done by showing that the smart card knows the private key. In both cases we also show that no replay attacks can occur.

We assume the following in both cases:

$$fresh\,(N)$$
$$B \text{ believes } fresh\,(N)$$
$$\xrightarrow{K} A$$
$$A \text{ believes } \xrightarrow{K} A$$
$$B \text{ believes } \xrightarrow{K} A$$

Case A

Claim: B believes A believes N

Proof: In the first step of the protocol we know that A sees N. After this we have:

$$\frac{\dfrac{B \text{ sees } \{N\}_{K^{-1}} \quad B \text{ believes } \xrightarrow{K} A}{B \text{ believes } A \text{ said } N} \quad B \text{ believes } fresh\,(N)}{B \text{ believes } A \text{ believes } N}$$

Which is what we wanted to show.

Case B

Claim: B believes A believes N, R

Assumptions: The smart card will always return the correct value of R.

Proof: In the first step of the protocol we know that A sees $\{N, P\}_K$. This gives us:

$$\frac{A \text{ sees } \{N, P\}_K \quad A \text{ believes } \xrightarrow{K} A}{A \text{ sees } N, P}$$

So A knows the password P, the nonce N and will now generate R based on the value of P. After A has sent its message we have:

$$\frac{\dfrac{B \text{ sees } \{N, R\}_{K^{-1}} \quad B \text{ believes } \xrightarrow{K} A}{B \text{ believes } A \text{ said } N, R} \quad \dfrac{B \text{ believes } fresh\,(N)}{B \text{ believes } fresh\,(N, R)}}{B \text{ believes } A \text{ believes } N, R}$$

Which is what we wanted to show.

Secure Spontaneous Device Association

Tim Kindberg and Kan Zhang

Hewlett-Packard Laboratories
1501 Page Mill Road, Palo Alto, CA 94304, USA
{timothy,kzhang}@hpl.hp.com

Abstract. One of the principal requirements of ubiquitous computing is support for spontaneous interoperation, in which two or more devices interact temporarily in ad hoc circumstances. Spontaneity typically arises when nomadic users carry one or more of the devices around. We contribute a method of using lasers for securing spontaneous device associations. The method provides users with physical evidence of which device they have connected their device to, as well as setting up a secure channel between the devices.

1 Introduction

One of the principal requirements of ubiquitous computing is support for spontaneous interoperation [1], in which two or more devices interact temporarily in ad hoc circumstances. Spontaneity typically arises when nomadic users carry one or more of the devices around. One example of spontaneous interoperation is where a user walks into a cybercafé with a personal digital assistant, phone, laptop or other personal mobile device, and accesses the café's printer. Another is where two users meet at a conference and exchange documents between their laptops. In a third example, teenagers in a shopping mall carry wireless multi-user game-playing devices. The teenagers pick one another and associate their devices in distinct groups corresponding to separate playing sessions.

Users can benefit greatly from interoperation with such spontaneously discovered services, but a challenging issue is security. Spontaneous associations take place over wireless networks, which by default are open to attack and hard for humans to deal with, since there is no physical evidence of what is connected to what. Users may inadvertently make connections to the wrong device. In many cases the associations take place in untrusted and unfamiliar environments. To return to our examples, the café user might trust the café's printer but doesn't necessarily trust other customers; and she doesn't want the document she sends to end up on someone else's laptop. The conference attendees have talked sufficiently to trust one another but don't want other attendees to access or tamper with the documents they exchange. The teenager in the mall trusts certain of their peers but not others to join their game.

A *secure association* is where each of two or more devices possesses the others' network addresses and the devices share a secret key. The devices can then encrypt their communications and use hash codes to guarantee the privacy and integrity of data transmitted between them. However, at a human level it is not enough for Fred Bloggs' device to guarantee to him that it has set up a secure association with "the

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 124–131, 2003.

device at network address 55.55.55.55". He needs to know that there is a secure association between his laptop (say) and that particular laptop over there in the hands of Mary Jones – and not a PDA in the pocket of someone at a nearby table. What we require is *physical validation* of an association, which means verifying the physical identity of the other party in an association. Physical validation can be seen as the physical counterpart of cryptographic authentication of identity.

Our contribution in this paper is to use lasers to achieve secure spontaneous associations between devices. We use them as physically constrained channels [2] so that humans can verify the validity of the associations they spontaneously make; we enable this functionality using only local, peer-to-peer communication and without trusted third parties.

The contribution can be seen as the wireless equivalent of a physical cable. For example, Fred could take a short physical cable from his pocket to connect his laptop to Mary Jones' laptop. Then neither would be in any doubt as to which two devices were connected and, if the cable had suitable properties, their communications would be secure. But Fred's data is still only secure if his trust in Mary proves well founded. Mary's device could maliciously publish his documents on the Web when it received them. We provide the same degree of security – and insecurity – but without wires.

2 Related Work

There are routine ways of securing an association over a wireless network but none is suitable for spontaneous association in unfamiliar circumstances. A Virtual Private Network connection (often used over 802.11) requires pre-registration and login. Users mutually authenticate their Bluetooth-connected devices by typing the same password on the devices, but exchanging a password securely out of band may be socially and physically awkward.

It is possible to construct a secure network discovery system [3] which provides certified responses to clients, but that begs the question of how to determine a certification authority in spontaneous circumstances. Moreover, even if the discovery system returns accurate information there may be devices of similar types but too little information to tell them apart. Suppose there are two printers in a location (A and B in Figure 1). How can the visitor know from their discovered names which printer is which? It will be inconvenient, at best, if her document appears at the wrong printer.

Several projects have suggested physical mechanisms to create associations. The 'resurrecting duckling' design for spontaneous networks [4] provides for 'secure transient associations'. The authors of that paper suggest that a key might be displayed on one of the devices but prefer 'physical' – electrical – contact as a means of key exchange. Those techniques achieve security but only at short range.

The Smart-Its project [5] introduced a method to establish an association between two handheld devices by holding them together and shaking them. Each device captures and broadcasts its own movement pattern. By matching the received movement patterns with its own, one of them can find the other and establish a connection between them. This protocol is subject to spoofing by an eavesdropping attacker.

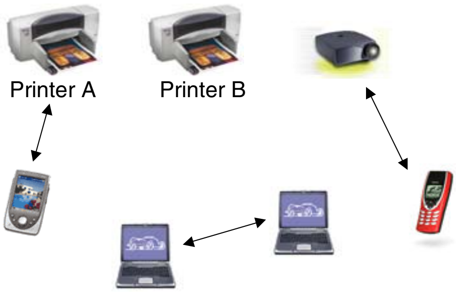


Fig. 1. Associations between personal and infrastructure devices.

It is preferable for devices to be securely associated at a more convenient distance. Feeney et al. [6] and Balfanz et al. [7] have suggested using a short-range infrared beacon on the target device. But precautions such as shielding would be required to protect the client from spurious infrared transmissions, whose source cannot be distinguished by distance and can only partially be distinguished by angle.

The SWAP-CA specification [8] for wireless networking in a home environment introduced what is commonly referred to as the two-button protocol, which was investigated more recently by Iwasaki et al. [9]. Users simultaneously trigger two devices into an 'association' mode, usually by pressing a button on each device. But the two-button protocol must use broadcast and is subject to timing-based attacks.

A 'physically constrained communication channel' [2] has the property that only certain principals with certain physical attributes may send or receive messages over the channel. The techniques we have been exploring involve validating associations using constraints based on location information and signal propagation characteristics.

Several groups (e.g. [10]) have investigated the combination of RF (radio frequency) and ultrasound signals to provide location information based on signal travel times. In earlier work, we devised methods for secure association based on those techniques [11]. That design has the valuable property that it serves for locating a device as well as securely associating with it. But there are drawbacks: the engineering difficulty of realising sufficiently accurate orientation and distance information; the added cost for each device; and the fact that the whole association procedure relies on a network discovery system.

A good example of a wireless channel with highly constrained signal propagation is a laser beam. Under conditions of appropriate engineering, only devices placed precisely at the two ends of a laser beam may communicate over it. While we are not aware of security-related uses, others have suggested the use of laser beams in pointing devices for ubiquitous systems [12, 13].

3 System Model

We consider two classes of devices that can interoperate spontaneously in a ubiquitous computing environment (Fig. 1) – without cables or prior software configuration:

1. *personal* devices such as laptops, personal digital assistants (PDAs) and mobile phones carried by nomadic users; these devices contain, in general, private data that needs securing.
2. *infrastructure* devices such as printers, which are more or less fixed in environments and ordinarily have no human operator; these devices are situated for public access – for our purposes, they are not themselves protected resources.

In each case we shall consider, a personal *initiator* device instigates an association with a *target* device, which may be an infrastructure device or another personal device. We assume that there is a human protocol, if need be, for deciding who shall be the instigator. If a group of devices is to be associated, we achieve this through multiple pair-wise associations.

The personal devices all have a means of wireless communication (e.g. 802.11 or Bluetooth), and they are assumed to be able to communicate with other personal devices or infrastructure devices over a wireless link or a combination of wireless and wired links.

By default we assume that the target device is in line of sight with the initiator – a reasonable assumption, since the user has established trust in the target device.

4 Secure Association Using Lasers

Our design for secure device association employs lasers, which provide relatively precise physically constrained channels. We equip the initiator device with a laser whose output can be rapidly switched to provide a data stream. The target device is equipped with a securely attached light sensor that can read the data emitted by the initiator device. A device that can play either role has both laser and sensor.

We assume no network discovery system. Rather, the user has identified the device she wants to associate with and is in line of sight with it. We first treat the case where the user wants to associate her personal device with an infrastructure device.

4.1 Validating Associations between Personal and Infrastructure Devices

The user presses the “lock-on” button on her personal device. That causes a laser beam to be emitted that pulses recognisably (so that she know she is not yet associating her device) but without data transmission. She orients the beam onto the target’s light sensor. Ringwald [12] discusses how the beam can be made wide enough to accommodate the user’s shaking hands but not so wide as to permit the beam to fall on the wrong device’s sensor. Moreover, the sensor can be designed with a suitable surround so that an attempt at placing a covert sensor would be obvious. Once the user is satisfied with the beam’s placement, she presses the “associate” button, at which point the two devices engage in the following protocol over two channels: the laser beam *L* and the network *W* (all or partially wireless) between the two devices.

The “Simple” Protocol

1. Initiator creates new session key Ks
2. Initiator → Target (over *L*): M1 = Initiator-address, Ks

3. Target→ Initiator (over *W*): M2 = {Initiator-address, Target-address, MAC}, where MAC is the message authentication code computed on (Initiator-address, Target-address) using key Ks.

When the initiator receives M2 in step 3, it verifies that the first value matches its own address in network *W* and the received MAC is correct. When successful, the two devices have secretly exchanged a session key Ks and their network addresses for their future communication. Moreover, the initiator's operator can be sure that she has set up a secure connection to the device at which she previously pointed her laser beam; any third party that had intercepted the laser beam would be so close to it as to be obvious to the human.

By definition, the target device in this scenario is open, so an attacker would be attempting to spoof the target device, or to obtain Ks to break the secrecy or integrity of communicated data; or it might try to deny service.

Spoofing and breakages of secrecy and integrity are impossible: by assumption the laser beam is well collimated so an attacker cannot obtain Ks. An attacker can, however, shine a spurious laser beam on the target device to deny service. If two beams land at more or less the same time on the same target device, then they might collide in their data transmission (which is detectable using checksums for laser packets).

On the other hand, if two users unwittingly shine their laser beams on the same target device together, then they can re-try if there is a collision; otherwise, they may concurrently access the target, as with any networked service.

4.2 Protocols between Personal devices

In this scenario, the goal is to associate two personal devices. Each of two users has a personal device equipped as in the previous protocol. Each personal device additionally has a light sensor attached, like the light sensor attached to the infrastructure device in the previous protocol.

The previous protocol validates a secure association to an infrastructure device for the human operator of a personal device. In personal-to-personal device association, both devices have human operators who require validation. We give two protocols for exchanging a session key between the devices.

The "Lights" Protocol

In this protocol, each personal device also has an omni-directional light (not a laser light) that is bright enough to be seen by both users. The first three steps of the protocol are identical to those of the "simple" protocol above.

1. Initiator creates new session key Ks
2. Initiator → Target (over *L*): M1 = Initiator-address, Ks
3. Target → Initiator (over *W*): M2 = {Initiator-address, Target-address, MAC}, where MAC is the message authentication code computed on (Initiator-address, Target-address) using key Ks.
4. Initiator (on receipt and verification of M2), blinks its light.

When the initiator receives M2, it verifies the message as in the simple protocol. If successful, it starts blinking the light that is visible to the user of the target device.

When the user of the initiator device sees the light blinking, she knows that the target device has received the secret key Ks. Moreover, when the user of the target device sees the light blinking, she knows that the initiator satisfactorily received M2. She thus can infer that the target has associated with the initiator, if she is sure that the initiator's user pointed the laser at the intended target.

This protocol is identical to the simple protocol in its first three steps and so has similar resilience to attacks. Since the attacker is unable to read the laser communication from the initiator, they do not know Ks and cannot forge the message needed to convince the initiator to turn on its light.

The "Symmetric" Protocol

There is some potential for confusion in the lights protocol if two concurrent runs occur between adjacent pairs of devices. Can one target-device's user be sure, on seeing a light blinking, that it was *this* device that initiated the association and not *that* one – which could also have shone its beam on the target at the same time? The following protocol enables us to associate two personal devices, without the potential ambiguity of the lights protocol but with more for the target device's user to do.

The initiator and target devices are both equipped with a laser and a sensor. However, unlike the previous protocol, this protocol doesn't require a blinking light.

1. Initiator generates a random number R1
2. Initiator → Target (over *L*): Initiator-address, R1
3. Target generates another random number R2
4. Target → Initiator (over *L*): Target-address, R2
5. Initiator → Target (over *W*): M1={R2}Ks, where Ks = hash(R1, R2), where hash() is a secure one-way hash function
6. Target → Initiator (over W): M2={R1}Ks

When the target receives message M1 in step 5, it decrypts the message using Ks and verifies that the decrypted value is equal to the value R2 it sent to the initiator in step 4. If the verification is successful, it knows that the initiator has successfully received R2 and hence the association with the initiator is validated. Otherwise, the target aborts the protocol. Similarly, when the initiator receives M2 in step 6, it decrypts the message and verifies that the decrypted value is equal to R1. If the verification is successful, the initiator is assured of the association with the target. Otherwise, the initiator aborts the protocol. When both verifications are successful, the initiator and target devices have agreed upon a session key Ks shared only between themselves. In other words, they have set up a validated and secure association.

By constructing the key Ks as hash(R1, R2), we make it difficult for either device to determine the session key in advance.

"Multi-device" Protocols

The preceding protocols can be adapted for setting up validated and secure associations among a group of personal devices, such as where a user aggregates visual and audio rendering devices for a multimedia playing session, or where a group of users play a game together. The goal is to transmit both a secret key and, to keep our description simple, a multicast address for the session along with that key.

We support two trust models. In the first, one of the devices in the group is identified as the leader for the purpose of setting up a secure association among the group. Choosing a leader is a social process rather than a technical one; any device can potentially become the leader. The leader can then generate a new session key Ks for the group and run steps 2 – 4 in the lights protocol with every other device in the group. In the end, every device in the group will have Ks and a validated and secure association has been set up among the group of devices, transitively through the leader. In the symmetric protocol both devices contribute to the resulting session key Ks. To enable a leader to choose and distribute a session key for a group, we can modify the symmetric protocol by letting Ks = R1 in step 5. As a result, the initiator can be the leader of the group and run steps 2 to 6 with every other devices in group to distribute the session key Ks and hence set up a validated and secure association among the group.

In the second trust model, a user may trust any user who already has the key – not just the leader. One of the devices in the group is chosen (again, by a social protocol) as the initiator, and it constructs a key and distributes the key by the operation of either the lights protocol or the modified version of the symmetric protocol. Thereafter, the same key can be distributed transitively through chains of users who acquire the key by the operation of the same protocol.

5 Discussion

We have described protocols for validating secure associations set up spontaneously between devices. The protocols enable nomadic users to securely associate their devices without communication with third parties. The users do so on the basis of dynamic judgements about the trustworthiness of individual devices in generally untrustworthy and unfamiliar environments.

The protocols substitute physical evidence for conventional certification (digital signatures from a trusted third party). The evidence derives from physically constrained communication channels [2], which in general are such that receipt of a message sent over them implies some physical property of the sending and/or receiving principals. Only *that* device could have sent a message to *this* device (in this case, over the laser beam).

Other types of constrained channels exist, including ones constructed from ultrasound [11] or infrared signals. But the laser-pointing protocol has the advantage of in-built precision and range. It also provides highly visible feedback to the user about the associations she makes.

It remains for us to implement our protocol. Existing implementations of laser-pointing devices on handheld devices such as Ringwald's [12] show that lasers and photo-voltaic cells can be simply interfaced to the serial port. There are health and safety considerations, but Class 2 lasers, with a power output of less than 1 mW, are safe to view directly for short periods of time. Our challenge is to evaluate the usability of our mechanisms, and we intend to do so both for associating to an infrastructure device, and the game-playing scenario.

References

1. Kindberg, T., and Fox, A.: System Software for Ubiquitous Computing. IEEE Pervasive Computing, vol. 1, no. 1 (2002), 70-81.
2. Kindberg, T., Zhang, K., and Shankar, N.: Context authentication using constrained channels. 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA), June 2002.
3. Czerwinski, S., Zhao, B., Hodes, T., Joseph, A., and Katz, R.: An Architecture for a Secure Service Discovery Service. Proc. fifth annual ACM/IEEE international conference on Mobile computing and networking (1999), 24-35.
4. Stajano, F., and Anderson, R.: The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks". In B. Christianson, B. Crispo and M. Roe (Eds.) Security Protocols. Proc. 7th International Workshop Proc, Springer-Verlag, (1999).
5. Holmquist, L.E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., and Gellersen, H.-W.: Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts. Proc. Ubicomp 2001, Springer-Verlag LNCS 2201 (2001), 273-291.
6. Feeney, L., Ahlgren, B., and Westerlund, A.: Spontaneous Networking: An Application-oriented Approach to Ad Hoc Networking. IEEE Communications Magazine vol. 39, no. 6, (2001).
7. Balfanz, D., Smetters, D.K., Stewart, P., and Wong, H.C.: Talking to strangers: authentication in ad-hoc wireless networks. Network and Distributed System Security Symposium; 2002 February 6-8; San Diego; CA; USA.
8. Shared Wireless Access Protocol (Cordless Access) Specification (SWAP-CA), Revision 1.0, The HomeRF Technical Committee, 17 December 1998.
9. Iwasaki, Y., Kawaguchi, N., and Inagaki, Y.: Touch-and-Connect: A Connection Request Framework for Ad-hoc Networks and the Pervasive Computing Environment. In Proc. IEEE PerCom (2003), 20-29.
10. Addlesee, M., Curwen, R., Hodges, S., Newman, J., Steggles, P., Ward, A., and Hopper, A.: Implementing a sentient computing system. IEEE Computer, vol. 34, no. 8 (2001), 50–56.
11. Kindberg, T., and Zhang, K: Validating and Securing Spontaneous Associations between Wireless Devices. To appear in Proc. 6th Information Security Conference (ISC'03), October 2003.
12. Ringwald, M.: Spontaneous Interaction with Everyday Devices Using a PDA. In Proc. Supporting Spontaneous Interaction in Ubiquitous Computing Settings, a workshop held at Ubicomp 2002. http://www.dcs.gla.ac.uk/~pd/Workshops/papers/ringwald.pdf.
13. Beigl, M.: Point & Click - Interaction in Smart Environments. In Proc. Handheld & Ubiquitous Computing, Springer Verlag, Lecture Notes in Computer Science, Vol. 1707 (1999) 311–314.

AwareCon: Situation Aware Context Communication

Michael Beigl, Albert Krohn, Tobias Zimmer, Christian Decker, and Philip Robinson

TecO, University of Karlsruhe, Vincenz-Priessnitz-Str. 1
76131 Karlsruhe, Germany
{michael,krohn,zimmer,cdecker,philip}@teco.edu

Abstract. Ubicomp environments impose tough constraints on networks, including immediate communication, low energy consumption, minimal maintenance and administration. With the AwareCon network, we address these challenges by prescribing an integrated architecture that differs from classical networking, as it features an awareness of the surrounding situation and context. In various settings, where AwareCon was implemented on tiny battery driven devices, we show that applications and usability of devices benefit from this approach.

1 Introduction

Communication between (small) embedded computing and sensing devices is an integral facility in Ubiquitous Computing scenarios. It is often the case that existing communication technologies, such as wireless LANs, mobile phone networks, or standards for wireless personal area networks such as Bluetooth or IrDA, form the communications backbone of these scenarios. However, it is our experience that these networks lack a significant measure of situation adaptability and are optimized for particular settings, not necessarily representative of typical Ubicomp application scenarios. For example, WLAN has increased in popularity as a standard for data transfer in offices and homes, while Bluetooth and IrDA are convenient replacements for cables between devices in close range. However, Ubiquitous Computing settings are characterized by a greater variance in networking ranging from sparse communication out in the field, where devices may come together infrequently, to dense communication environments consisting of possibly hundreds of devices in one room.

In this paper we propose a communication system that is designed to adapt to various settings by generating and using context and situational information for improving the overall system performance. In Ubicomp, context or sensor data is often used as input for applications, which run as a layer on top of a network stack like in TinyOS [1] and ContextCube [2]. Our design and implementation of an Aware-of-Context (AwareCon) – network, shows how a network could benefit from intrinsic processing of context information rather than simply acting as its transport medium.

Existing networks for small, embedded devices often emphasize particular (context) implementation issues. For example, the Prototype Embedded Network (PEN) [4] provides advanced energy saving by using very simple and passive devices. Blue-

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 132–139, 2003.

tooth [5] and other standard bodies in the domain of ad-hoc networking (e.g. IEEE 802.15 [6] (esp. TG2) and ZigBee [7]) currently have no built-in notion of context awareness in their protocol stacks.

Today, the development of context aware systems such as Toolkit [8] and TEA [9], build on top of the communication subsystems rather than allowing a measure of integration. Our contribution with AwareCon functions as a bridge between situation aware systems and network design in Ubicomp.

In Ubicomp settings, an assortment of devices works together to unobtrusively support the human through response to explicit human computer interaction, and, more significantly, rapidly changing context. We concur with other researchers (e.g. [10]) that unobtrusiveness is a core operational feature of many technical systems that are intended for background environmental augmentation. Further challenges include low power consumption (e.g. [11]), immediate communication between unknown devices, robustness and no system administration (e.g. challenges 2,3,6 for Ubicomp in homes from [12]). From our experience with systems and settings requiring non-stop operation (e.g. MediaCup[13], AwareOffice [14] with several years of operation), we found frequent maintenance to be particularly inconvenient, and suggest that minimization of the maintenance effort is an essential and practical requirement for Ubicomp devices.

2 Context and Situation Aware Networking

Meeting the above challenges with strictly classical networking approaches has proven to be an arduous undertaking. We therefore claim that *situation awareness is essential for Ubicomp devices and their communication, to meet acceptable performance efficiency*. This concept underpins the AwareCon communication system. Each instance of the network (realized as small, mobile devices) must be able to produce, store and consume relevant information.

In situation aware networking we use context to represent situations. The situation of an artefact (a mobile device) is a collection of context information that leads to adaptive decisions, including communications behavior. Context in Ubicomp settings is derived from environmental information, obtained from sensor data, and from meta-information of the application domain, generated by a *context producer*. Context models like the one proposed by Dey, Abowd and Salber in [15], which focus on how to handle the "content" of the context, do not cope with its underlying structure. In contrast, we concentrated on the structural attributes of context and identified validity, relevance, reliability and context history as the most influential properties. They define the representation of context in AwareCon. Although AwareCon is implemented as a complete network protocol stack, it has unique features for context aware communication. Context information is contained within the payload of communications packets. Nevertheless, it is still possible to transport any other raw data on higher protocol layers.

Context is thus used in two ways in the proposed system: Firstly, context data for use by applications is encoded within the payload of communications packets. Secondly, situational context information used for network optimization is processed at separate component of the protocol stack.

3 The AwareCon Network Stack Architecture

The AwareCon protocol stack provides typical network features for coding, access and transport. The AwareCon protocol is divided into layers and components. This structural approach is a well-founded principle in network engineering.

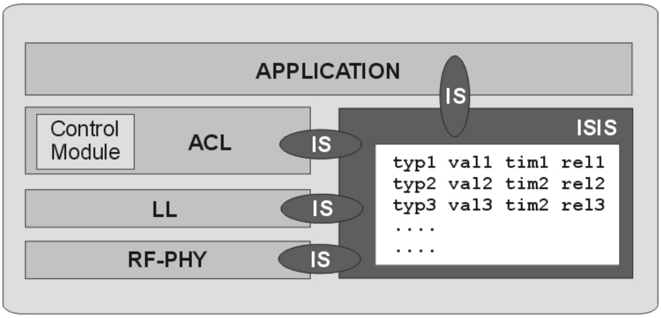


Fig. 1. AwareCon Protocol Architecture

The basic architecture consists of 5 components (Figure 1). Four of the components are traditional communication layers: the Radio Frequency Physical Layer (RF-PHY), Link Layer (LL), and Application Convergence Layer (ACL), with a control module for remote administration and the Application Layer. In addition to the traditional network layers, we have introduced a component called the Internal Situation Store (**ISIS**), with accompanying Interpretation Stubs (**IS**) at each of the traditional layers. The ISIS is the core, facilitating component for context awareness as it holds all information relevant to the context-based enhancement of the communication and application. Contexts can originate from internal or external sources. Internal sources include functions that interpret status variables of the protocol stack or sensor values, whereas external sources are e.g. remote devices or service points that broadcast context information. Data elements stored in the ISIS are clearly separated from the actual payload data that is transported for application purposes.

In the style of the earlier identified attributes of context, we decided to implement the following 4 attributes for contexts stored in the ISIS: *type* of situation, *value* of situation, *time* stamp of last change and *reliability* of the value. The Interpretation Stubs are used to push situation data (context) of interest into the ISIS, and subsequently to interpret outbound data for the respective part of the system and provide an easy access interface. Theoretically, any context and situational information could be stored in the ISIS. At the current stage, the generation and storage of the following contexts are implemented:

- energy resources (battery level)
- processor load (percentage of busy time)
- link quality (bit error rate/packet error rate)
- number of active devices

These values are produced and consumed for improvement of the performance of the network stack and applications. In the next section, we take a closer look at three of the mentioned context values in the ISIS. We explain where they are generated and how the consumption of that information improves the network or application.

4 Applications Based on AwareCon

For some time now we have constructed various Ubiquitous Computing environments and applications, which have been intermittently running for several years. The AwareCon network is an outcome of collective findings and experience made with these settings. At the same time, we also use the settings as test-beds for evaluating the functional features of the AwareCon. AwareCon is implemented on TecO's generic hardware platform (see Figure 2) designed and built during the Smart-Its project [16]. Smart-Its are very small computing, sensing and communication devices that are especially designed for post hoc augmentation of everyday objects. In the following sections we also refer to Smart-Its as artefacts.

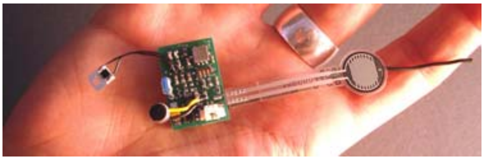


Fig. 2. TecO's Smart-Its: A general sensing, computing and communication platform

4.1 Energy Saving for AwareOffice

Long-term operation of computerized artefacts is important on the Ubicomp agenda, as was mentioned at the beginning of this paper as one of the main network challenges. AwareCon exploits context awareness as a means of *energy saving*.

The "chair appliance" in the AwareOffice scenario is a meeting room chair with a Smart-It device attached. These were our primary implementation and evaluation test beds for the power saving mechanisms. In this setting, two contexts stored in the ISIS inform the energy saving process: *the number of active devices* - a context produced in the network stack (RF Layer) and the *remaining energy* – a local sensor value available on each mobile device. The chair appliance interprets certain movements of a chair to resolve that someone is sitting or that the chair is being moved. The detected status information is then communicated to other artefacts in the environment via the AwareCon network. The chair shows typical communication behavior for electronic artefacts in Ubicomp: the communication is sparse, and unpredictable. The transmission of outbound packets results only from user interaction.

Chairs – here as example of everyday objects equipped with electronics – are typically unsupervised and are seldom or never maintained, such that some consideration is necessary to achieve long-term operation of the electronics (reduce mean power consumption). Consequently, the major contributor to energy consumption was the communication. Without human interaction, the chair's Smart-It is in sleep mode, consuming minimal power. Upon movement of the chair, the attached Smart-It starts its processor and communication, sends out a packet containing the actual movement state or pattern and returns to sleep mode. Running the processor and protocol stack consumes around 100 times more power than the sleep mode. Therefore it is a clear goal to minimize the "up time" of the chair's Smart-It. The mean time for sending an outbound packet has a significant impact on the over all mean power consumption. The longer the send operation has to wait for channel access, the more energy has to be spent waiting with the processor and protocol stack running. An arbitration in the Link Layer resolves the distributed transmit inquiries. The mean delay time until a packet is transmitted depends on the number of active, continuously transmitting devices. In this circumstance, context awareness suggests a prioritization of only those artefacts (in this case the chair appliance) that are low on energy resources, to reduce the delay for sending of outbound packets. Assuming various types of artefacts in one scenario, only those with critical energy resources (e.g. depending on batteries) will invoke the mentioned energy saving mechanism. To control the collision rate, the knowledge of the number of active devices is necessary to adjust the arbitration for the channel access.

With a mean of 1% power-up quota (99% sleep mode) for the chair's Smart-It, around 100 days of operation can be achieved with a 500mAh battery. Giving the outbound packets of the chair a higher priority could easily shorten the mean delay for packet delivery by a factor of ten. This results in a longer lifetime of around 200 days (ignoring the self discharge of the battery).

A predictable lifetime of an artefact until replacement or recharge of batteries is another issue. As stated, the power consumption of the chair's Smart-It depends strongly on the activity that takes place with the chair. This will result in different battery cycles due to different activities. However, for administration and maintenance it's important and helpful if no artefact ever runs out of energy before the predicted lifetime.

Therefore, the network protocol stack can react according to the *energy resources* context of the ISIS. In a situation where Smart-Its are low on remaining energy resources, they can reduce their transmit power to save energy and invoke a simple repeat request (flooding) of their packets to transport them further. While transmitting, the transmit power is more than 50% of the whole power consumption of the artefact. Transmitting with minimum power and a 1% power-up quota can reduce the mean power consumption by 30%, implying 30% longer lifetime after starting the mechanism. In doing so, Smart-Its use energy of the surrounding devices – functioning as transmission repeaters - to equilibrate the energy consumption of artefacts in one area.

4.2 Number of Active Devices: Application Triggering

Some applications depend on partner devices in a certain area. The absence of these partner devices would shut down the applications. RELATE, a 2D surface location system for tangible objects [17], is one example built by us using Smart-Its as communication technology. RELATE enables the distributed localization of objects e.g. on a white board with no infrastructure, and is therefore an excellent Ubicomp example scenario.

The ISIS value *number of active devices* (produced in the RF Layer) is used to determine if possible partners are around. The RELATE technology uses infrared signals to determine location. The scanning and localization of objects requires certain effort in computation and energy. Therefore, it is essential to avoid invoking the location mechanisms when no partner devices are around.

With the knowledge of the *number of active devices,* it is further possible to influence the update rates of the localization algorithm. This is of special interest for RELATE because the human interaction with objects carrying RELATE functionality should be supported and enabled in real time, which makes scaling and update rates critical issues.

4.3 Computation Time Prediction for Context-as-a-Key

Small embedded devices like Smart-Its and Smart Dust [3] are often single processor solutions. This reduces the complexity and the energy consumption, and simplifies the development process in the research stage. Computation time spent for the networking on such a single processor design can consume a substantial part of the overall available computation time. This may lead to conflicts with certain application tasks - e.g. digital signal processing - that are also in need of great amounts of processing power. In real time applications computation has to be finished after a certain time frame. With high network activity, this real time behavior might not be reached.

The AwareCon stack provides the context *processor load*. This information – produced in the RF layer – can be used by the application to control its own run time behavior. *Context-as-a-Key* [18], a security service for mobile devices, provides an encrypted communication based on symmetric keys generated from a common context. Devices in close range use synchronously sampled audio data from their environment to generate these keys. The key generation and encryption – running on one processor with the network stack - demands high computation effort. Depending on the known *processor load,* the security service is able to predict its response time in advance and the application can calculate the possible frequency of secure communication. This enables adaptive behavior of time critical applications.

5 Needed Technical Characteristics of the AwareCon Stack

As shown in the above applications, it is necessary to produce certain context values to implement the context awareness of the network system. These values must be generated reliably and fast. The context awareness seriously depends on the performance of the production of the necessary contexts. AwareCon provides certain features to generate several contexts quickly and dependably. For effective situation aware communication, we summarize and amend necessary properties and features that are implemented in the AwareCon system (especially the protocol stack) on Smart-Its:

- Decentralized and cellular architecture and media access
- Ad-hoc behavior and spontaneous book-in into a network
- Real time communication and synchronization
- Predictable processor load due to the network stack
- Low power consumption
- Small package
- Easy (no) administration and maintenance

The network protocol stack addresses the typical requirements for an Ubicomp setting by providing real ad-hoc link establishment (typical 12ms), low power consumption (<10mA), various energy saving mechanisms, permanent synchronization (4us between any pair of nodes), distributed access control with predictable and adjustable collision rate, error control, over the-air configuration and programming, and addressing, yet also anonymity. Including these features, the fixed slotted AwareCon protocol achieves a data-rate of 48kBit/s in the 868Mhz band.

6 Conclusion

The features and functionalities explained position AwareCon as an example of a network architecture appropriate for typical Ubicomp scenarios. It uses situation aware communication in order to address earlier identified challenges and also enables applications to gain from the situation awareness. The examples in the previous sections showed how energy saving and application control during run time is possible with an underlying subsystem that generates context awareness. The selected values stored in the ISIS were influenced by the application and the effort to implement. These are therefore not the extent of selectable attributes and require some further assessment. Further context values in the ISIS and new ways to produce and consume them are necessary. The ISIS is meant to be a design initiative, advancing the beneficial influence of situation awareness on applications and their supporting infrastructure (hardware, networks etc.). Context aware applications can use the ISIS in order to maximize performance potential. With the AwareCon architecture, context information is available for any component of the system and opens a new playground for adaptive algorithms and applications.

References

1. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D.E., Pister, K.:System Architecture Directions for Networked Sensors. ACM Sigplan notices,v. 35(#11) pp. 93-104, Nov (2000)

2. Hähner, J., Becker, C., Bauer, M., Schiele, G.: ContextCube - Providing Context Information Ubiquitously.International Workshop for Smart Appliances and Wearable Computing (IWSAWC) 2003, Providence, RI USA, Mai (2003).
3. Pister, K.S.J., Kahn, J.M., Boser, B.E.: Smart dust: Wireless networks of millimeter-scale sensor nodes. University of California, Berkeley, USA (1998)
4. Bennett F., Clarke D., Evans J.B., Hopper A., Jones A., Leask D.: Piconet: Embedded Mobile Networking, IEEE Personal Communications, Vol. 4, No. 5 (October 1997) 8-15
5. Sonnerstam, D. (Hrsg.): Specification of the Bluetooth System Version 1.0 B, Bluetooth SIG (1999)
6. 802.15, Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs), 14 June (2002)
7. ZigBee Alliance [Online] from: http://www.zigbee.org [Accessed 04/22/03]
8. Dey, A.K., Salber, D., Abowd, G. D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. HCI, Vol. 16 (2-4) (2001)
9. Gellersen, H.W., Beigl, M., Schmidt, A.: Sensor-based Context-Awareness for Situated Computing. SEWPC00, Limerick, Ireland (June 2000)
10. LaMarca, A., Brunette, W., Koizumi, D., Lease, M., Stefan Sigurdsson, S., Sikorski, K., Fox, D., Borriello, G.: PlantCare: An Investigation in Practical Ubiquitous Systems. UbiComp 2002, (2002)
11. Smit, G.J.M, Havinga, P.J.M: Lessons learned from the design of a mobile multimedia system in the MOBY DICK project. In Proceedings of HUC2k, Bristol, UK (2000)
12. Edwards, W.K. and R.E. Grinter: At Home with Ubiquitous Computing: Seven Challenges" Proceedings of the UbiComp 2001, Atlanta, Georgia, USA, 2001
13. Beigl, M. Gellersen, H.W., Schmidt, A.: MediaCups: Experience with Design and Use of Computer-Augmented Everyday Objects. Computer Networks, Vol. 35, No. 4 (2001)
14. The AwareOffice Initiative at TecO [Online] TecO, University of Karlsruhe, Germany, Available from: http://www.teco.edu/awareoffice [Accessed 04/22/03]
15. Dey, A., Abowd, G., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. HCI, Vol. 16, (2001)
16. Beigl, M., Zimmer, T., Krohn A., Decker, C., Robinson, P.: Smart-Its - Communication and Sensing Technology for UbiComp Environments. Technical Report ISSN 1432-7864 (2003)
17. Assessment of relative positioning technologies for compositional tangible interfaces (RELATE), [available online] http://www.teco.edu/relate/
18. Robison, P., Beigl, M.: Trust Context Spaces: An Infrastructure for Pervasive Security in Context Aware Environments. Proceeding of SPC 2003, (2003)

liquid: Context-Aware Distributed Queries

Jeffrey Heer, Alan Newberger, Chris Beckmann, and Jason I. Hong

Group for User Interface Research, Computer Science Division
University of California, Berkeley
{jheer,alann,beckmann,jasonh}@cs.berkeley.edu

Abstract. As low-level architectural support for context-aware computing matures, we are ready to explore more general and powerful means of accessing context data. Information required by a context-aware application may be partitioned by any number of physical, organizational, or privacy boundaries. This suggests the need for mechanisms by which applications can issue context-sensitive queries without having to explicitly manage the complex storage layout and access policies of the underlying data. To address this need, we have developed liquid, a prototype query service that supports distributed, continuous query processing of context data. This paper articulates the current need for such systems, describes the design of the liquid system, and presents both a room-awareness application and notification service demonstrating its functionality.

1 Introduction

One important aspect of the evolving ubiquitous computing vision is the development of context-aware computing [11], in which sensor networks and other data sources are leveraged to provide computing systems with an increased awareness of a user's physical and social environment. Sensed and inferred context data can then be exploited to provide enhanced computing services [5].

Until recently, most technical work on context-aware computing has focused on low-level architectural support [5,9,14], and has been primarily concerned with the acquisition and storage of context data. For example, in Dey et al.'s Context Toolkit (CTK), context widgets are associated with sensors and used to collect and aggregate sensed data [5]. Client applications can then subscribe to these widgets over the network to access and monitor context data. Hong's Context Fabric (Confab) [9] provides the abstraction of information spaces (*infospaces*), logical storage units using the tuple space paradigm [6] that serve as context repositories for individual entities in an environment (e.g., people, locations, and objects). Similar to the CTK, Confab users can set up subscriptions to specific infospaces to monitor changes in context data (e.g., subscribing to a room infospace to monitor its current occupants).

Although this work greatly supports the acquisition and organization of context, the means of monitoring context can be much improved. Unfortunately, the nature of context data provides some inherent challenges to achieving this. First, we expect context data to be distributed—organized by a number of logical and physical separations, including organizational and privacy boundaries.

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 140–148, 2003.

Second, context is dynamic—objects are moved around, people walk in and out of rooms, temperatures fluctuate, and activities begin and end.

Consider a scenario in which a student is busy working on a paper and needs urgent feedback from their advisor. The student could use a context-aware notification service to raise an alert when their advisor is in the same building and interruptible [10]. Supporting such applications requires systems capable of monitoring and dynamically re-routing connections to reflect changes in the physical world.

Existing context-aware architectures support these requirements, providing distributed storage and subscription services for context data, but they still place an unnecessary burden on context-aware application developers. Using current architectures, the previous notification scenario would require an application to manage a host of subscriptions to remote repositories, maintain multiple network connections, and perform all intermediate data processing. Duplicating this level of work for all desired context-aware applications is simply unacceptable. Those systems that do attempt to simplify context access, namely the CTK's Situation Abstraction [5], do so at the cost of centralizing all related context data, limiting scalability. What is needed are general, adaptive, and decentralized infrastructure services that both scale to wide-level deployment and simplify application design. As a result, we believe a distributed query service for context data is the logical next step for context-aware infrastructures.

Many of the technologies needed for creating such a service have been well investigated in the database literature. Distributed databases (e.g., [15,16]) support the separation of data across multiple nodes in a network. Recent work in streaming databases [2,3,4,13] breaks the relatively static model of traditional systems, viewing data as a (possibly infinite) sequence of items, and can thus monitor data that changes over time. Finally, semi-structured databases (e.g., [12]) illustrate techniques for dealing with data of varying and dynamic structure. Although each of these individual areas have been well researched, the needs of context-aware systems lie in their intersection, which unfortunately has yet to be fully realized in the database community.

To address these needs, we built liquid, a query service that supports context-aware applications. Building on emerging work in the field of streaming databases, liquid supports distributed, decentralized query processing over continuously changing context data, stringing queries across various context repositories, re-routing queries as changing context dictates, and providing streaming results back to query issuers. liquid not only collects results from distributed context repositories, but distributes the actual *processing* of the query across the network, decentralizing query processing and improving scalability. As such, liquid helps to fill a current void in the design space of both database systems and context-aware computing infrastructures.

2 liquid Design and Implementation

The liquid query service is built atop an existing context-aware storage infrastructure, the Context Fabric (Confab) [9]. In this section we describe the design of Confab as relevant to liquid, and present the design of liquid itself.

2.1 Context Fabric

The Context Fabric is a distributed context-aware infrastructure with services to support the acquisition and retrieval of context data. Using Confab, people, places and things (*entities*) are assigned network-addressable logical storage units called Information Spaces (*infospaces*) that store context data. Sources of context data, such as sensor networks, can populate infospaces to make their data available for use and retrieval. Applications retrieve and manipulate infospace data to accomplish context-aware tasks. Infospaces provide an abstraction with which to model and control access to context data. Infospaces can be both distributed across a network or managed on a centralized server. We expect some combination of both to be typical of a Confab deployment. Confab is implemented in Java, using HTTP for network communication.

The basic unit of storage in an infospace is the *context tuple* (tuple). Tuples may contain arbitrary data, including descriptions of other entities related to the original, containing infospace. Elements common to all tuples are *type*, a textual name describing the relationship of a tuple to the containing infospace's entity; one or more *values*, each identified by name; and an optional *entity-link* denoting the address of an infospace for an entity described by the tuple. Such tuples' entity-link elements refer to the infospace of the other entity. For instance, the infospace for a specific room may contain numerous tuples of type 'occupant', each with values denoting a name and email of an occupant of the room and an entity-link referring to the infospace that hold tuples on behalf of that occupant. Context tuples are represented externally as XML documents, and are the basic data unit processed by liquid queries.

2.2 liquid

liquid is a query processing service, implemented in Java, that deploys on top of the Context Fabric. The processing of a liquid query involves receipt of the query over the network, translation of the received query into a query plan, execution of that plan (including, if necessary, the dispatch of sub-queries to remote query processing units), and the return of any results to the query issuer. Both queries and query results are represented in XML and communicated over HTTP.

A core concept of liquid is the entity type path, which is illustrated in Figure 1. A type path specifies a tuple to be retrieved at the end of a sequence of infospaces, each infospace being specified by its type relation to the infospace prior in the sequence. For example, to get the occupants of the room you're currently in, the entity type path of this data, relative to your own personal infospace, would be "location.occupant". This scheme, similar in spirit to type

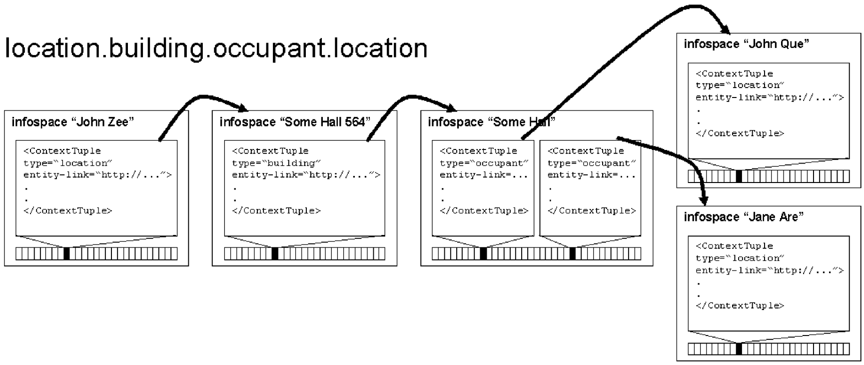


Fig. 1. The evaluation of an entity type path relative to a root infospace. This path returns the specific locations of people in the same building as the query issuer.

paths used in semi-structured databases like [12], specifies data across a logically separate and possibly distributed set of storage repositories.

liquid queries are issued in a declarative format (currently in XML) which specifies the tuples to be retrieved as a set of type paths. This format organizes the query into partitions that can be serviced at individual infospaces. For instance, a query retrieval path of "location.occupant.age", designed to retrieve the ages of people in the same room as the query issuer, would be split into 3 partitions: one retrieving "age" tuples, which is contained within another partition retrieving "occupant.age" tuples, which is in turn contained in a partition retrieving "location.occupant.age" tuples. Each partition can also contain conditions that tuples in the retrieval path must satisfy. When queries are received by a liquid node, the system determines which partition must be executed locally, and uses it to generate a *query plan*. A query plan is a structured tree of *operators* that perform query processing and partition forwarding to remote infospaces. liquid operators use the standard iterator model of query processing. Interested readers are referred to [7] for an introduction to these concepts.

Query execution consists of the management and evaluation of the generated query plans. Queries are first registered with the *query execution manager*, which holds a registry of live queries and oversees query scheduling. Upon receiving a query, the execution manager associates the query with its own *executor*, an object which oversees the evaluation of a single query, requesting new results from the query operator tree and returning them to the query issuer. The executor returns results to an issuer through a callback provided by liquid's network communication layer. Our current implementation uses a single-thread of execution per query, which the execution manager oversees using a thread pool.

The basic unit of query processing passed between operators is the *result item*. The result item is a time-stamped collection of context tuples, each indexed by their entity type path. Result items also maintain a status flag that indicates if an item has been *inserted*, *deleted*, *updated*, *exited*, or *expired*. *Inserted* or

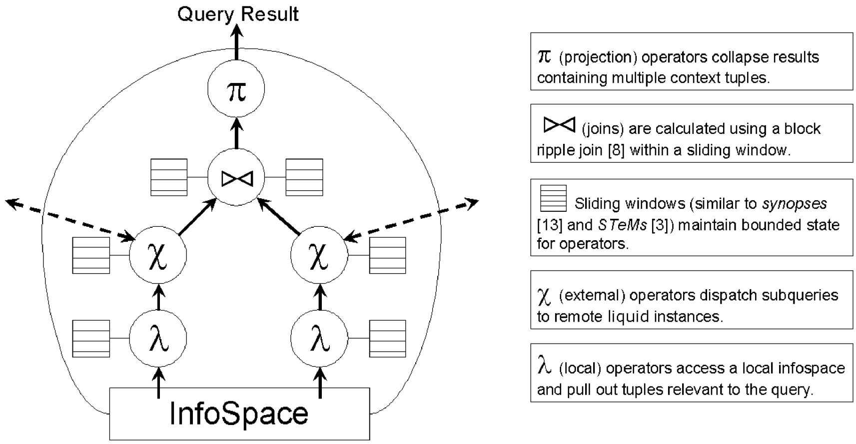


Fig. 2. An example liquid query operator tree. Dotted arrows indicate remote queries.

deleted items have either been added or removed from the scope of the query (e.g., a person walks into a monitored room, and then walks out). *Updated* and *exited* items are those whose content has been updated, either in such a way that the item still meets the query conditions (*updated*) or that it no longer does (*exited*). Finally, *expired* items are those that the system can no longer track due to bounded resources (see the discussion on windows below).

2.3 Operators and Windows

Currently, liquid supports the traditional database operators of selection (σ), projection (π), and joins ($\bowtie$). We also introduce two new operators specific to liquid: local (λ) operators, which are responsible for interfacing to a local infospace, and external (χ) operators, which are responsible for managing and dispatching remote sub-queries to other liquid nodes. The operator tree also features windows, caches which keep a history of tuples while maintaining bounded operator state. Figure 2 shows an example of an actual liquid operator tree.

Windows provide operator state within bounded resources, akin to synopses in [13] and state modules (SteMs) in [2]. liquid windows support result item insertion and removal, item searching, and automatic item eviction. Windows issue callbacks to their associated operators to inform them of eviction events. This functionality is used to keep query issuers apprised of which results are being actively monitored and which have *expired*.

liquid employs a number of operators to service queries. Local (λ) operators are responsible for retrieving context data from underlying infospaces. The λ operator monitors the local infospace for tuple insertions, deletions, and updates. A λ operator also maintains a window to keep a history of retrieved tuples, primarily to handle *exited* tuples. Items evicted from the window are flagged as *expired* and added to the queue of items to be passed up the operator tree.

External (χ) operators retrieve data from remote infospaces by querying other liquid nodes. Result items are pulled from a child operator, and the entity-links of these items are used to issue queries to liquid instances at other infospaces. The query results are then streamed back, enqueued into the χ operator through a callback from the network layer. The operator keeps a registry of running queries, indexed by the result item that spawned it. *Updated* items arriving from the child operator may cause running queries to be canceled and then reissued as necessary. *Deleted*, *exited*, and *expired* items cause the corresponding queries to be canceled. Accordingly, χ operators maintain their own window, allowing the results from canceled queries to be effectively "recalled" by passing the proper *deleted*, *exited*, or *expired* items up the tree.

Selection (σ) operators filter items based on a given condition, while projection operators (π) collapse results items by removing unnecessary context tuples. These operators are stateless and hence quite easy to implement. Join ($\bowtie$) operators evaluate two incoming streams and merge the contents of two result items if they meet a specified condition. Each stream has an associated window, over which the join is computed using a variant of the block-ripple join [8]. Join operators have also been designed to carefully manage the cases when *deleted*, *exited*, and *expired* items must be propagated up the operator tree. This results in the desirable property that query issuers are guaranteed to be notified whenever a result item goes "stale".

3 Applications

As a first evaluation of the liquid system and its impact on context-aware application design, we've built a simulation system and two applications showcasing the system's functionality. We are in the process of setting up instrumented spaces in which to explore real-world context-aware scenarios, but in the meantime we have built a location simulator to support system testing and application design. The system provides a complete model of our building on our campus, and represents any number of people and objects located within the building. The system allows users to simulate movement of people and objects through a web-based interface (Figure 3), or can be programmed to automatically simulate activity. The simulator is a surrogate for an underlying sensing infrastructure, populating and updating infospaces as simulated context changes over time.

3.1 Application 1: Co-occupant Awareness

We built the first application to provide augmented awareness within a room or space. For a given person, the application provides the e-mails and webpages (if available and allowed by privacy policies) for all people currently in the same room as the query issuer. As people enter and exit the room, this result is automatically updated, and as the query issuer moves from place to place, the query is automatically re-routed to the correct infospace for the new location. This application showcases liquid's ability to both handle distributed data (i.e.,

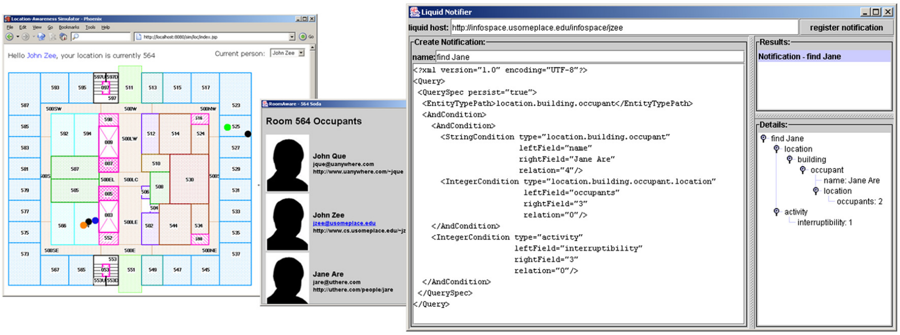


Fig. 3. Application screenshots of the location simulator (left), the room awareness application (center), and the notification service interface (right).

across infospaces for both people and locations) and respond to the dynamic nature of context. The user interface for this application is shown in Figure 3.

3.2 Application 2: Notification Service

Next, we built a context-aware notification service, in the spirit of CybreMinder [5], that allows the user to specify conditions of which they would like to be notified. This specification is performed using the interface in Figure 3, which currently shows a running query for the "advisor hunting" scenario mentioned in the introduction. The application issues user-provided queries and provides notification to the user as query results arrive. The processing of the notification conditions is completely provided by the liquid infrastructure (e.g., by computing joins), leaving the application developer responsible only for generating the appropriate queries and displaying the results. The evaluation of these queries is dispersed throughout the infrastructure, supporting scalability and reuse of computation through both caching and sub-plan sharing.

4 Future Work and Conclusion

Despite the success of our system, there is still much work to be done. First, our modeling scheme of context data is preliminary. We are in the process of determining the right balance between simplicity and expressive power in how context-aware architectures model the surrounding world. Next, a more natural query language is needed to further simplify the burden of application designers. Privacy issues with context data are also a serious concern. While access control is handled by the underlying Context Fabric, liquid will need to consider authentication and encryption in the evaluation of queries. Another major area is optimization and evaluation, including query scheduling and parallelism, sub-plan sharing, window size adjustments [13], and dynamic operator optimization [1,2]. An important benefit of incorporating query processing into context-aware computing systems is the potential for improved efficiency of the infrastructure, allowing informed resource management and avoiding replicated work. It is our hope that by identifying these system needs, the ubicomp and database fields might inform each other, providing both new challenges and application domains for database researchers and new systems and techniques for the designers of ubiquitous computing environments.

In conclusion, this paper presented liquid, a continuous query processing engine for supporting context-aware applications. To this aim, liquid supports persistent, streaming queries over distributed data repositories, and dynamic query re-routing in response to changing context. Current context-aware computing architectures do not efficiently support these services, often relying on context-aware application designers to implement them on their own. liquid hopes to simplify and enhance the creation of context-aware applications by moving these services into the available infrastructure.

Acknowledgments

We are indebted to the guidance and insights of Anind Dey, Eric Brewer, James Landay, and Jennifer Mankoff. This work was supported by an NSF ITR for context-aware computing. The first author was supported by an NDSEG fellowship.

References

1. R. Avnur and J. M. Hellerstein. “Eddies: Continuously Adaptive Query Processing.” In SIGMOD 2000.
2. S. Chandrasekaran, et al. “TelegraphCQ: Continuous Dataflow Processing for an Uncertain World.” CIDR 2003.
3. J. Chen, D. DeWitt, F. Tian, and Y. Wang. “NiagaraCQ: A Scalable Continuous Query System for Internet Databases.” In SIGMOD (2000).
4. M. Cherniack, H. Balakrishnan, M. Balazinska, D. Carney, U. Cetintemel, Y. Xing, and S. Zdonik. “Scalable Distributed Stream Processing.” CIDR 2003.
5. A. K. Dey. “Providing Architectural Support for Building Context-Aware Applications.” Ph.D. thesis, December 2000, College of Computing, Georgia Institute of Technology.
6. D. Gelernter. “Generative communication in Linda.” ACM Transactions on Programming Languages and Systems. 7(1), 1985.
7. G. Graefe. “Query Evaluation Techniques for Large Databases.” ACM Computing Surveys 25 (2): 73-170, 1993.
8. P. J. Haas and J. M. Hellerstein. “Ripple joins for online aggregation.” In A. Delis, C. Faloutsos, and S. Ghandeharizadeh, editors, SIGMOD 1999.
9. J. I. Hong. The Context Fabric. http://guir.berkeley.edu/cfabric.
10. S. E. Hudson, J. Fogarty, et al. “Predicting Human Interruptibility with Sensors: A Wizard of Oz Feasibility Study.” CHI 2003.
11. Human-Computer Interaction Journal: Special Issue on Context-Aware Computing, Vol. 16, No. 2-4, 2001.

12. J. McHugh, et al. “Lore: A Database Management System for Semistructured Data.” SIGMOD Record, 26(3):54-66, September 1997.
13. R. Motwani, et al. “Query Processing, Resource Management, and Approximation in a Data Stream Management System.” CIDR 2003.
14. B. Schilit. “System architecture for context-aware mobile computing.” Unpublished doctoral dissertation, Columbia University, 1995.
15. M. Stonebraker, et al. “Mariposa: a wide-area distributed database system.” VLDB Journal 5, 1 (Jan. 1996), p. 48-63.
16. R. Williams, et al. “R*: An Overview of the Architecture.” Technical Report RJ3325, IBM Research Lab, San Jose, CA, December 1981.

Is Context-Aware Computing Taking Control away from the User? Three Levels of Interactivity Examined

Louise Barkhuus[1] and Anind Dey[2]

[1] The IT University of Copenhagen, Glentevej 67
2400 Copenhagen, Denmark
barkhuus@it.edu

[2] Intel Research Berkeley, 2150 Shattuck Avenue
Berkeley, CA 94704, USA
anind@intel-research.net

Abstract. Context-aware computing promises a smooth interaction between humans and technology but few studies have been conducted with regards to how autonomously an application should perform. After defining three levels of interactivity between a mobile computing device and its user: personalization, passive context-awareness and active context-awareness, we test which approach will limit users' perceived sense of control. We also investigate users' preferences for the three approaches. We conducted an experimental case study, using mobile phone applications to exemplify the three levels of interactivity. Our study shows that users feel less in control when using either passive or active context-aware applications than when personalizing their own applications. Despite this we also find that context-aware applications are preferred over the personalization oriented ones. We conclude that people are willing to give up partial control if the reward in usefulness is great enough.

1 Introduction

While context-aware computing aims to facilitate a smooth interaction between humans and technology, few studies of how users perceive context-aware interaction have been performed. Most research focuses on the development of technologies for context-awareness as well as the design of context-aware applications. Example applications are numerous and the level of interactivity within these varies greatly, ranging from letting the user manually define parameters on how an application should behave, to automatically providing the user with services and information that the developer finds relevant. Here we present a study whose goal is to examine users' sense of control and their preference to interactivity level. Because mobile telephony is the most widely employed ubiquitous computing device, we are using this as an example of context-aware computing.

We define context-aware computing as 'an application's ability to detect and react to environment variables' [1]. Here we define three levels of interactivity for

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 149–156, 2003.

context-aware applications: personalization, passive context-awareness and active context-awareness. *Personalization* is where applications let the user specify his own settings for how the application should behave in a given situation; *passive context-awareness* presents updated context or sensor information to the user but lets the user decide how to change the application behavior, where *active context-awareness* autonomously changes the application behavior according to the sensed information [3]. Drawing on the three types of interaction, it is our belief that users feel a loss of control when using passive or active context-aware applications but not when they personalize their mobile device, and prefer personalization despite the higher interaction cost.

We present a case study that analyzes users' attitudes towards each of the three levels of interactivity. Focusing on early stage analysis, we evaluate the services before actually implementing them; with this, we hope to gain insight into users' reactions and use-level as early in the development process as possible. We assign each participant with a specific level of interactivity and trace their user habits for five consecutive days, in order to understand their potential use habits and preferences. It is our goal to obtain results that will guide the development of future context-aware services in having an appropriate level of interactivity.

Our study found that users' sense of control decreases when autonomy of the service increases, as suggested by previous research [4]. We believed that personalization would be preferred and would be more accepted than both passive and active context-awareness, however, the results of our study do not support this. Instead we find that people prefer context-aware applications over personalization oriented ones.

In this paper we first discuss the three levels of interactivity and review relevant literature. Second we present the method used in our case study. Third, we report the results and fourth we provide a discussion of the findings. Finally, we conclude and provide suggestions for further research.

2 Three Levels of Interactivity

In this section we review related work and although we define three levels of interactivity, we review active and passive context-awareness levels together. They are closely related and, unlike personalization, are both based on sensor information.

2.1 Personalization

Personalization, sometimes also referred to as customization and tailoring, is a common feature of computing applications.Personalization of desktop applications is a widely researched area [7,8,10]. Researchers argue that the diversity and dynamics of applications call for an increased level of tailoring in software, and that this emphasis on customized functionality will add to the the user experience and smoothness of interaction [10]. Limiting the scope to mobile computing, it is exemplified by the settings in a mobile phone, defining the user's preference for background picture and ringing profiles. One interesting finding is that, even though many desktop applications, as well as larger websites, offer personal tailoring, the majority of users use the default setting or change a small subset of the possible features [6].

No studies focus directly on personalization within mobile computing. Some studies approach the subject with respect to users' personal attachment to their mobile phone [11], but no study has looked into users' preferences or perception of tailoring their handheld device. However, because mobile devices are inherently personal, it is likely that these users, in particular, will enjoy the advantages since the tailoring will not affect other users.

2.2 Active and Passive Context-Awareness

Since the notion of context-aware computing was introduced by Schilit et al. in 1994 [9], several definitions have been offered, often describing different levels of interactivity. Cheverst et al. for example, investigate whether information should be pushed towards the user or the user should be left to pull the information on his own in context-aware systems [4], whereas other researchers consider only push based applications to be context-aware [5]. In this study we draw on Chen and Kotz's definition of active and passive context-awareness [3]. Active context-awareness describes applications that, on the basis of sensor data, change their content autonomously, where passive context-aware applications merely present the updated context to the user and let the user specify how the application should change, if at all. A simple example of an active context-aware application is the mobile phone that changes its time automatically when the phone enters a new time zone. In the corresponding passive context-aware application, the mobile phone prompts the user with information about the time zone change and lets the user choose whether the time should be updated or not.

While many researchers differentiate between the levels of interactivity, they rarely agree on where to separate them. Cheverst et al.'s 'push' approach is described in the same terms as our definition of passive context-awareness, while their pull approach falls in a category between personalization and passive context-awareness [4]. Another distinction is provided by Brown and Jones, who define the levels of 'interactive' and 'proactive' [2]. Interactive applications cover our definitions of both personalization and passive context-awareness where proactive is defined almost identically to active context-awareness. None of this research however, considers the difference in users' perception of the different levels. The three levels of interactivity presented here serve as the basis for our case study, which we will describe in the next section.

3 Case Study

Our study is conducted as an experimental case study comparing users' responses towards applications, representing our three levels of interactivity. It is based on a five-day fill-in diary, which are supplemented with qualitative interviews with

Table 1. The three levels of services presented to participants.

Service	**Personalization**	**Passive Context-Awareness**	**Active Context-Awareness**
A: Private ringing profiles	Different ringing profiles that are set manually	The phone prompts the user to adjust the profile when sensing it is in a meeting or class	The phone automatically changes profile when sensing the user is at a meeting or in class
B: Public ringing profiles	Different ringing profiles that are set manually	The phone prompts the user to adjust the profile when sensing it is in a movie theater or at a restaurant	The phone automatically changes profile when sensing the user is at a movie theater or at a restaurant
C: Lunch service	Manual search for appropriate lunch place	Single alert around noon for lunch place according to users' preferences	Alerts the user when passing by a lunch place of relevance and suggests places at noon
D: Class slides	Manual search to see if class slides are available online	If signed up, the phone alerts user of available slides for class	Automatic alert every time the teacher updates class slide website
E: Location tracking	Manually location tracking of predefined friends	Locations tracking of friends and setting to alert when they are within a certain range	Location detection of friends that alerts when they are within 300 feet of user
F: Activity tracking	Display of potential call-receiver's social situation (e.g. meeting, home, out)	In a new context, the phone prompts the user to display the user's situation to possible callers	Automatic switch to display of social situation when entering a new context

a subset of the participants. By studying participants' reactions and attitudes towards context-aware applications, it is our goal to obtain results that will guide the development of future context-aware services in having an appropriate level of interactivity.

3.1 Research Method

The study is designed as a between-subjects study, where each participant is assigned to a group within one of the three levels of interactivity. Because it was

Table 2. General participant demographics.

N=23	**Personalization**	**Passive Context-Awareness**	**Active Context-Awareness**
N	8	8	7
Average age	23.7	22.9	25
Average mobile phone ownership	2.2 years	2.6 years	2.7 years
Average user level (a scale from 1–6)	3.1	3.8	3.4

not possible to implement all the services within each interaction level, we introduced participants to new services that they were to 'pretend' were available on their mobile phones. The applications are described in table 1. At the end of each day (for 5 days), the 23 participants filled in a journal of how many times they would have used the services and to what degree they thought the services would have been useful. The services' usefulness and level of intrusion were evaluated on a scale from 1-5 and the journal form left room for additional comments. To supplement the journal, qualitative interviews were conducted with 6 of the participants to elaborate on their reactions to the interactive services and their overall perception of context-awareness within mobile computing.

Six different services were proposed and each was described in terms of personalization, passive context-awareness as well as active context-awareness. The participants were presented with services that belonged to one level of interactivity and not informed that there were different groups. The journals were filled out over the same 5-day period.

3.2 Participants

23 participants were selected using the criteria that they should have a mobile phone and use it frequently, at least three times a day on average. Both students and non-students were recruited with an age range from 19 to 35 and average age being 23.7. 10 out of the 23 were students and their mobile phone ownership ranged between 1/2 and 6 years, with an average of 2.5 years. The participants were randomly assigned to the groups, resulting in a slight difference in average age and use level among the groups. However, the values are fairly close and this difference is therefore not considered in our analysis; see details of each group in table 2. We now present the findings of the case study and conclude on the participants' sense of control and preferences.

4 Perception of Control

Our hypothesis that people felt less in control when using context-aware services, than when personalizing applications, was found to be true. Users' perception of control was measured in three ways. First, the participants were asked directly if they felt a loss of control over their mobile phone when 'using' the services; second each of the services were evaluated according to this perception at the end of each day, and finally the interviews were analyzed.

From the direct questions we found a correlation between the given level of interactivity and the participants perception of control of -0.26, meaning that the more autonomous the service is, the less users felt in control. This is not statistically significant at the .025 level. However, when considering the two levels of context-awareness as one category, the correlation is -0.31, which is statistically significant at the level of .025. When evaluating each of the services in relation to sense of control, the participants' results indicate that the personalization group felt more in control than both other groups of context-awareness for services A, B and D, but the opposite was true for service E. The latter result is perhaps due to the rather controversial nature of tracking the user's location; it does not matter if display of location information can be controlled, since for some people it is an uncomfortable feature. The last two services did not show any difference in perception of control. Finally the interviews indicated that most of the participants feel they have control of their mobile phone with their currently available personalization-based applications, but several of them worried about this control when they were introduced to context-aware features.

5 Preference for Active and Passive Context-Awareness

We found that preference for interactivity level actually contradicted our initial hypothesis. Participants preferred active context-awareness and passive context-awareness over personalization.

The preference for different levels of interactivity was measured in two ways. First each of the services was assessed according to how many times the participant would have used it on a specific day, and, second, the participants were asked to rate the services for their usefulness.

We found an unexpected statistically significant correlation between the levels of interactivity and use for two of the services (B with a correlation of 0.29 and E with a correlation of 0.31, both significant at the .025 level), meaning that the higher level of interaction resulted in higher level of preference. Personalization-based services were used the least, whereas active context-aware services were used much more. Service F on the other hand had a statistically significant correlation of -0.32 (significant at the .025 level); the most popular version of this sevice was the personalization oriented and the least preferred was the active context-aware one. The rest of the services did not show any correlation between preference for interactivity level at all.

An interesting result is that preference for interaction level did not vary across individuals, but varied across service. Hence, some services were very popular

(A, B, E), where other services were regarded as fairly irrelevant(service D [1] and F), meaning that they were used rarely, and overall were too intrusive (service C). As a general finding the active context-awareness approach was preferred. Even when taking into account the fact that some participants are high level users, defined as 7 calls a day or more, and some are low level users, defined as 4 calls a day or less, the finding that active context-awareness is slightly preferred is still consistent.

Although some of the results were not what we expected, they provide interesting insight into users' perception of control and preference for interactivity levels, which we will elaborate on in the discussion.

6 Discussion

The finding that participants felt they had less control in the context-aware groups but still preferred the context-aware approaches, might at first seem contradictory. However, it should be considered that owning a mobile phone in itself constitutes some lack of control since the user can be reached anywhere at anytime; the user might have less control, but are aware that this is the cost of becoming more interactive and in achieving a smoother everyday experience.

Although our study results provide support for highly interactive applications for mobile computing, by indicating that people would use them to a fairly high degree, the applications should still be developed with caution. The incurred cost due to loss of control can result in users turning off a service. While the participants initially liked many of the active context-aware services, they might become frustrated by their perceived lack of control and eventually turn the service off.

Lastly it should be noted that even though participants were 'equipped' with a highly interactive mobile phone for the duration of the study, imaginary approaches like this are not always sufficient to tell if users would actually behave as they self-reported. One commonly observed factor is that even though potential users may disregard a technology a priori, they may adopt it anyway for various reasons. Examples include text messages over the phone where reasons to adopt this communication form can be peer pressure or change of attitude due to a realization of the value it provides.

7 Conclusions and Further Work

In this study, we have examined peoples' sense of control and preference for three levels of interactivity within mobile computing. The study shows that participants feel a lack of control when using the more autonomous interactivity approaches but that they still prefer active and passive context-aware features over personalization oriented applications in most cases. Our conclusion is that

[1] The results from the class slide service were adjusted to account for the participants who are not students.

users are willing to accept a large degree of autonomy from applications as long as the application's usefulness is greater than the cost of limited control.

Because our study is a theoretical evaluation of three levels of interactivity, the logical next step is to develop actual context-aware and personalization oriented services. They should be user tested for the same parameters as this study and compared to how people rated the virtual services. To complement our diary and interviews, observational methods should be used to more accurately determine how users handle highly interactive applications. Finally, it should be noted that for rigor, future studies should account for participants' mobile phone experience and use level.

References

1. L. Barkhuus. Context information vs. sensor information: A model for categorizing context in context-aware mobile computing. In *Symposium on Collaborative Technologies and Systems*, pages 127–133, San Diego, CA, 2003.
2. P. J. Brown and G. J. F. Jones. Context-aware retrieval: Exploring a new environment for information retrieval and information filtering. *Personal and Ubiquitous Computing*, 5(4):253–263, 2001.
3. G. Chen and D. Kotz. A survey of context-aware mobile computing research. Paper TR2000-381, Department of Computer Science, Darthmouth College, November 2000.
4. K. Cheverst, K. Mitchell, and N. Davies. Investigating context-aware information push vs. information pull to tourists. In *Proceedings of Mobile HCI 01*, 2001.
5. T. Erickson. Ask not for whom the cell phone tolls: Some problems with the notion of context-aware computing. *Communications of the ACM*, 2001.
6. S. R. Page, T. J. Johnsgard, U. Albert, and C. D. Allen. User customization of a word processor. In *Proceedings of CHI '96*, pages 340–346, 1996.
7. T. Rist and P. Brandmeier. Customizing graphics for tiny displays of mobile devices. *Personal and Ubiquitous Computing*, 6(4):260–268, 2002.
8. G. Rossi, D. Schwabe, and R. Guimarães. Designing personalized web applications. In *Proceedings of the tenth international conference on World Wide Web*, pages 275–284, 2001.
9. B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proceedings of the 1st International Workshop on Mobile Computing Systems and Applications*, 1994.
10. O. Stiermerling, H. K., and V. Wulf. How to make software softer - designing tailorable applications. In *Symposium on Designing Interactive Systems*, pages 365–376, 1997.
11. A.S. Taylor and R. Harper. Age-old practices in the 'new world': A study of gift-giving between teenage mobile phone users. In *Proceedings of CHI 2002*, volume 4, pages 439–446, 2002.

Tools for Studying Behavior and Technology in Natural Settings

Stephen S. Intille, Emmanuel Munguia Tapia, John Rondoni, Jennifer Beaudin, Chuck Kukla, Sitij Agarwal, Ling Bao, and Kent Larson

Massachusetts Institute of Technology
1 Cambridge Center, 4FL
Cambridge, MA 02142 USA
intille@mit.edu

Abstract. Three tools for acquiring data about people, their behavior, and their use of technology in natural settings are described: (1) a context-aware experience sampling tool, (2) a ubiquitous sensing system that detects environmental changes, and (3) an image-based experience sampling system. We discuss how these tools provide researchers with a flexible toolkit for collecting data on activity in homes and workplaces, particularly when used in combination. We outline several ongoing studies to illustrate the versatility of these tools. Two of the tools are currently available to other researchers to use.

1 Introduction

Products developed and tested in laboratories often fail when introduced into natural settings such as homes and workplaces. Human behavior in these natural settings is strongly tailored to the settings themselves and to the behavior of the people nearby. Differences between expectations of how people will behave and how they actually do behave in the complexity of real settings contribute to product failures. Developers often make erroneous assumptions about the need for, use of, and reaction to new technologies.

Ubiquitous computing technologies are more intimately tied to their environments and the people within those environments than desktop applications. Therefore, these ubicomp applications must respect, detect, and respond to the interaction between the person and the natural setting. The common desktop model of software evaluation with usability labs will fail to capture the influence of setting on behavior. Simulation of realistic natural behavior in the laboratory is difficult, because to do so requires reconstructing the environments themselves.

Ubicomp developers, therefore, face a challenge: how can models of behavior in natural environments be obtained? Once applications are created, how can they be evaluated *in situ* and over long stretches of time? Effective development and deployment of persistent, context-aware interfaces will require tools and strategies for using those tools that can help researchers establish good answers to such questions. Longitudinal study of interface use in natural settings is particularly challenging. Today, many ubicomp interfaces are proposed that would

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 157–174, 2003.

run for months or years at a time when deployed. Yet these systems are typically only tested for hours at a time in labs. Both the *impact of the environment* and the *impact of time* on the behavior of the users of these applications must be considered.

In this paper we describe three tools we have created, each in different stages of development. The first tool, context-aware experience sampling, extends electronic experience sampling to include a set of sensors that both collect data and proactively trigger data collection. The second tool is a data collection system of small, simple state-change sensors that can be quickly installed throughout nearly any environment to collect information about patterns of activity. These two tools, both of which are available to the research community, are most powerful when used in combination. We describe how we are deploying them in multiple studies to gather data on activity in natural settings. Finally, we describe a laboratory prototype of a third tool that further extends the observational capabilities of researchers and developers: image-based experience sampling.

2 Studying People in Natural Settings

Developing meaningful ubiquitous computing applications should begin with a thorough understanding of how people behave in their environments, how they perform the tasks to be undertaken, and how this behavior may be influenced by the introduction of new technologies. In short, first we must understand the behavior of people and *then* develop the technology. For instance, despite little evidence that non-gadget-oriented consumers want or need automated home lighting and HVAC control or remote control of appliances, many popular visions of ubiquitous computing technology advocate the use of complex sensing to achieve these goals (e.g. [22,8]). Often, proposed applications would strip control from end users and “simplify” life by acting as autonomous, controlling agents. Many demonstration applications are built upon unrealistic assumptions about user tolerance for applications that make erroneous control decisions.

2.1 Motivation: Empowerment vs. Control

An alternative approach is to develop ubiquitous computing environments that use technology not to automatically control the environment but instead to help users learn how to control the environments themselves. In this vision, many ubiquitous computing applications will not make a decision for the users but instead present information to users. The task of interpreting a suggestion or information in context rests with the user. Studies in preventive medicine and energy management have shown that simple, passive, but context-dependent and relevant indicators can have a dramatic influence on behavior (see [15]).

This approach has several advantages over a controlling approach:

– Information can be presented so that the occupant can react to without interrupting ongoing activity in potentially irritating ways; this is especially true

if information can be digitally "augmented" onto the physical environment itself using projection.
- Leaving the user in control of making decisions without confusing the occupant; the occupant will naturally consider contexts that the ubicomp system has not and adjust his or her actions accordingly.
- Algorithms that make suggestions can degrade gracefully; algorithms that make decisions typically do not.
- Lack of control over aspects of life has been shown to diminish health [24]; this strategy empowers the occupant.

A shift in focus from creating automatic ("smart") environments to environments that help the occupant learn how to take self control impacts not only the type of technology one might design but also one's outlook on how to conduct research to evaluate the work. A "Jetsonian" model of a future environment, for example, where the computer simplifies life with automated control, can be evaluated by designing demonstration environments in a traditional laboratory with little user involvement. However, environments that are designed to help occupants learn cannot be evaluated independently of the people using them. We need to study the *people using the technology* in realistic, non-laboratory settings for long periods of time and then measure whether our interventions are leading to learning and behavior change. This is the primary motivation for the development of the three tools described below for naturalistic observation.

2.2 Standard Approaches to Naturalistic Observation

Developers of ubiquitous and mobile computing applications for the home and workplace currently lack a powerful and economical assessment tool set. There are five classes of methods used to elicit user needs:

Interviews. Interviews are performed individually or using focus groups and are particularly effective for critique of an existing idea or (with an effective interviewer) gathering general information about user tasks. Often, however, users know more than they say in a single or even several interviews [25] and will tend to have difficulty understanding and recalling how context impacts their behavior (i.e. exhibiting selective recall and selective reporting biases [26]). One special form of interviewing consists of participatory design games that are conducted in the setting where the technology will ultimately be used.

Direct observation. Direct observation with trained observers does not suffer from selective recall and is considered the "gold standard" for assessment in medical and psychological research studying behavior in natural settings. Although direct observation can provide helpful qualitative and quantitative measures, it is costly, time-consuming, and disruptive and therefore not practical for many design tasks, particularly those where researchers need to invade private settings such as the home. Recently, direct observation with photographic and video analysis has been used (e.g. [14]).

Self-report: recall surveys. Despite the widespread use of self-report surveys for assessment of behavior in naturalistic settings, these surveys are known to suffer from recall and selective reporting biases - users can often not remember what they did or do not report what they actually did. Further, they often report what they did incorrectly [26].

Self-report: time diaries. To minimize selective recall and selective reporting bias, time diaries can be used. Users write down what they do during the day as they do it or at regular intervals [23]. Diaries can provide less biased data than recall surveys but are burdensome for the user, can impact the activity itself, and often result in records with missing information. Providing users with devices such as dictaphones, cameras, and video cameras has been used to simplify self-reporting.

ESM/EMA. The experience sampling method (ESM), also known as ecological momentary assessment (EMA), uses a timing device to trigger self-report diary entries [11,26]. In electronic ESM, survey questions can be answered on a portable computing device that "samples" (e.g. via a beep) for information. Only recently has ESM been employed for interface design [16,17,13,10,14]. Sampling can occur using fixed, statistical, or user-initiated schedules. With a sufficient number of subjects and samples, a statistical model of behavior can be generated. The ESM is less susceptible to subject recall errors than other self-report feedback elicitation methods [11,26], but at high sampling rates it can interrupt activities of interest and irritate subjects, resulting in some subject-selection bias [11].

3 Challenges: Improving Observation Tools

We are interested in using and developing sensing tools to improve the assessment techniques. We have created three such tools: environmental sensors (ES), context-aware experience sampling (CAES), and image-based experience sampling (IBES). No tool will suit every need, but ideally combinations of tools can be paired to achieve the properties discussed below:

Dense measurement of activity. For many in-home studies, researchers need a dense description of activity recorded from the home environment to analyze. The collection tool, however, should not impact behavior in the environment. Therefore, while sometimes invasive cameras or microphones can be used, in other cases sensors are needed that allow a researcher to study activity without direct, invasive observation. (ES)

Fast install/removal. Most homes and workplaces do not easily accommodate even the simplest of new technologies. Therefore, researchers require tools that can be used to study natural settings that can be retrofitted into these environments easily and at low cost. (ES, CAES)

Low cost. Both low-cost manufacturing and installation and maintenance of sensors must be possible so that they can be deployed in multiple households for long time periods. (ES, CAES)

Robust longitudinal data collection. Many studies require data collection over the course of weeks from home environments. (ES, CAES, IBES).

Longitudinal acceptability. Directly querying a user is a powerful technique, but one that should be done sparingly to avoid annoying a user (e.g. perhaps when an application is first used to perform some customization to the user's routines). We are experimenting with the combination of *context-aware* sampling and other tools to minimize the burden of self-report on subjects so the technique can be used for longer time periods. (ES, CAES, IBES)

Not perceived as invasive. To deploy technologies into homes for any extended period of time, they must not be perceived as invasive. (ES, CAES)

Autonomous operation possible. Although some tools require human intervention, ideally observational tools will be useful not only for observation but also for intervention with observation, where some event is detected automatically, some information is provided, and the user's response is observed. (ES, CAES)

Sensing technology can be used to extend and compliment traditional observational techniques in a cost-effective manner. Our goal is to select a suite of sensors that permits a researcher and, eventually an algorithm, to quickly analyze the data and construct a record of activity. Exactly what the researcher is looking for depends upon the activity. Therefore, the tools should be flexible. We advocate using a combination of room-mounted, body-mounted, and object-mounted sensors in combination with a self-report mobile computer tool. In the remainder of this paper we describe the three three tools we have created and some observations resulting from their deployment.

4 Tool 1: Context-Aware Experience Sampling

The development of the first tool began when members of our research team identified a need for a robust self-report data collection tool. As we worked on the development of computational perception algorithms for automatic identification of activity and contextual information from sensor data acquired from mobile computing devices, we realized that we needed accurate annotation of the sensor readings with the target activities to both train and test pattern recognition algorithms. In the laboratory obtaining this annotated data is a straightforward process: use direct observation and label activity in real time or through observation of video sequences. In our case, however, we were interested in identifying activities as people went about their lives outside of the lab. Direct observation was too costly, time-consuming, and invasive. Therefore, we employed self-report time diaries. Subjects were asked to keep diaries of their physical activity (e.g. walking to class, climbing stairs) and when they did it.

We encountered selective reporting and forgetfulness typical of self-report (see Sect. 2.2). Therefore, we decided to use electronic ESM to ease the subject's burden, improve the accuracy of time-stamps acquired, and reduce the data entry and coding burden of the experimenter. We found that a few commercial experience sampling programs are available (see listing from [6]) as well as one open-source program for old versions of the Palm and Windows CE operating systems [5]. However, in addition to lacking some new ESM functionality we desired and only operating on outdated hardware, the available software did not permit both the acquisition of user self-report data and the simultaneous acquisition of other data streams such as position (from GPS) and heart rate (from a wireless monitor). We therefore have developed new experience sampling software. The basic tool offers options not currently available in any other open-source or commercial sampling package (e.g. leaving an audio or photo annotation). In addition, the software can collect readings from sensor devices attached to the PDA.

4.1 Using Context to Trigger Self-report

In addition to extended data collection capabilities, the tool also provides one fundamentally new type of functionality: *context-aware* experience sampling. This feature permits researchers to acquire feedback from users only in particular situations that are detected by sensors connected to a mobile computing device. The context (location, time, event, biosensor data) can trigger the sampling.

Typically a researcher using experience sampling has three options: (1) sampling on a fixed interval schedule, such as every 30 minutes, (2) sampling on a random interval schedule, such as on average once every 30 minutes or sometime randomly within every 2 hour window, and (3) sampling in response to user initiative, where the user is told to make a data entry whenever performing a particular activity [26].

Our software has been developed in a modular fashion that allows new context-sensing sensors and software to be plugged in. These sensors permit researchers to use context-sensitive sampling where specific questions are asked only when a user does a specific thing (e.g. is near the store, which is determined by a GPS and map module, or has a change in heart rate, which is determined by a wireless heart rate monitor). Context-awareness modules permit a researcher to acquire more information about the behavior or situation of interest by sampling only during or just after the activities of interest. This minimizes the interruption annoyance of the ESM technique without compromising the quality of data acquired about the target phenomena.

4.2 Implementation

We have used a participatory design process to create the interaction model for the ESM tool so that it suits both researcher and subject needs. Our goal has been to create software that permits a device to be handed to a subject at the start of a week with only a few minutes of instruction and then to be returned a week later with question and sensor data. The interface has been designed so that it is self-instructing and easily understood by non-computer users.

The first release of the software includes capabilities for standard multiple choice question experience sampling using a time-sampled protocol. The software, written in C++, runs on the PocketPC platform and has been developed

(a) (b) (c)

Fig. 1. (a) The context-aware experience sampling software runs on standard PDAs and offers a variety of options for acquiring self-report data from users or subjects in experiments. This image shows one screen from sampling protocol being used by a study collecting training data on activities in the home setting that is being used to develop algorithms to detect everyday activities from sensor data automatically. (b) The software permits the use of data collection sensors such as cameras, microphones, and wireless heart rate monitor. This image shows a user taking a picture with the camera plug-in and one of the built in tutorial screens. (c) The software can also either prompt for sensor data such as taking a GPS reading or use continuously-acquired sensor data to trigger a set of self-report questions to be asked. Here a user is acquiring a GPS sample, which is done by holding up a small antenna plugged into the GPS device.

primarily for iPAQ devices due to the large number of hardware connection options these devices provide and the bright, high-contrast screen that makes the questions and instructions easier to read.

Researchers load a new protocol by modifying a comma-delimited text file. Our software also includes protocol development flexibility not found in other open-source ESM software [5]. These include options for (1) chaining complex sequences of questions based upon particular question responses, (2) aggregation of questions to minimize user disruption in some situations, (3) study suppression during some events, (4) multiple choice and multiple response questions, (5) manual specification of precise query times for particular questions or question sequences, (6) flexible question recurrence patterns (by weeks, days, hours, minutes), and (7) bounded randomization (min/max time to next query). Further, researchers can allow users to leave answers via audio recording or, if the PDA has a camera plug-in, answer a question by taking a picture. The tool can therefore be used to combine the powerful techniques of ESM and photographic analysis. The device includes just-in-time tutorials to aid subjects with some of the advanced functionality. Figure 1 shows screen shots of users answering questions, including questions that ask for data samples (picture, GPS) taken using plug-in hardware.

The context-aware functionality of the first release includes the ability to sample based not only upon the standard time-based protocol but also upon a subject's location, as obtained by a GPS plug-in. Therefore, researchers can design studies that sample only when near a location of known interest. Similarly, a wireless Polar heart rate monitor can be used to trigger samples based upon significant changes in heart rate.

Additional context-aware extensions are currently being added to the software, including the ability to sample based upon *particular activities* that have been detected using accelerometers, the GPS, and the heart rate monitors (e.g. "walking"). We are building an open source development community and invite researchers interested in using experience sampling or developing extensions to the context-aware experience sampling tool to visit the project pages found on SourceForge [6] and join the effort.

The primary benefit of context-aware experience sampling versus traditional experience sampling is that specific activities can be targeted. This means that intensity of sampling can be increased around those moments or activities being studied and reduced at other times. Further, the type of questions answered can be changed based upon the context detected by the software. Finally, context-aware experience sampling can be setup so that samples are delayed until after an activity of interest may have just finished rather than sampling at a random time during the activity, which may disrupt the activity itself.

5 Tool 2: Ubiquitous Environment State-Change Sensor System

We have developed a second observational tool that *passively* collects data via measurement of objects in the environment and compliments the self-report data collected by the context-aware experience sampling device.

We have created the software and hardware for a system of simple, robust sensors that can be ubiquitously installed into complex environments such as real homes and workplaces to collect data about activity. Analysis of the data, either by hand or by machine, may enable better understanding of activity in naturalistic settings and create new opportunities for development of context-aware applications.

In prior work where sensors have been ubiquitously installed into home or workplace environments, typically only a small number of sensors have been used or the studies have been conducted in relatively controlled research settings such as homes of the researchers themselves or close affiliates (e.g. [22,3,1]). The sensor installation itself is often difficult and time consuming to accomplish and troublesome to maintain.

Our system uses "tape on" sensors that can be quickly installed in a complex natural environment to measure just a few or several hundred object states, depending upon how many are used. A small team of researchers can install the system in a small, one-bedroom apartment of typical clutter and complexity in only about 3 hours. The devices have been used to continuously and passively collected data for two-week blocks in multiple subject homes. These subjects have had no affiliation with our research project.

Fig. 2. The second tool is a system of environment state change sensors that can be installed ubiquitously throughout an environment such as a home. Each device consists of a data collection board and a small sensor, and the components are literally taped to objects and surfaces for the duration of the data collection period. These images show the data collection board outside of the protective case and 11 of the 85 sensors installed throughout one subject's home.

5.1 Implementation

Our design goals were to permit several hundred low-cost sensors to be installed in an environment for at least two weeks, left unattended, and then recovered with synchronized data. Figure 2 shows a sensor device, which consists of the sensor itself connected by a thin wire to a 27mm x 38mm x 12mm data collection board. The devices are robust and easy to work with. Each fits snugly in a plastic case of dimensions 37mm x 44mm x 14mm. They use either reed switches, which are activated when brought into contact with a small magnet, or piezoelectric switches, which detect movement of a small plastic strip. Use of temperature, vibration, and load sensors is also possible. The plastic cases are literally taped onto surfaces using a non-damaging adhesive selected based upon the material of the application surface. The sensor components (e.g. reed and magnet) and wire are then taped to the surface so that contact is measured. Figure 2 shows 11 of 85 sensors that were installed in the home of one subject. They can be attached to many devices in the home, including light switches, containers, and furniture. A trained researcher can typically install and test a sensor in less than 3 minutes. The sensors in this subject's home operated for 16 days.

To save memory and cut cost, the boards save data with time stamps that have 1s resolution. To achieve well-synchronized measurements a temperature-compensated crystal oscillator is used to drive the real time clock of the data collection board. This achieves a time accuracy of +/- 1 minute per year if operated from 0 to +40C. To further improve the synchronization, prior to installation all of the boards are synchronized in a single session with a reference PC clock. When the data collection boards are recovered, the signals from each board are linearly interpolated to better match the reference clock. In boards installed in our laboratory, we have measured the synchronization after this correction to be +/- 2 seconds. The boards can record up to 3 activations and deactivations per second and can record a total of 2666 events in memory.

The total cost for parts and fabrication (in quantities of 150) for each data collection board as of February, 2003 is $24.66, where an additional $2 is required for each sensor (e.g. magnetic reed). When sensors are installed, each data collection board (which has a unique ID) is marked on a plan-view of the environment so that when the sensor data is collected the location and function of each sensor is known. The sensors are in continuous use at this time. We are sequentially installing and then removing them in different subject homes. The complete hardware and software specifications for the devices are available on request.

Although other portable sensing systems for ubiquitous computing applications have been developed [12,18,19,21], we are unaware of work where 100+ of these devices have been rapidly deployed in multiple homes as "tape on and forget" devices and collected data with non-researcher occupants in normal homes for weeks at a time. Several groups have hard-wired cabinets and mats [1] and a some kitchen appliances [3], but these systems have not been deployed throughout multiple, entire homes because of the difficulty of installation and maintenance. The simplicity and small size of our data collection boards make it possible to cost-effectively deploy large numbers in non-laboratory settings. Although our boards are simpler than distributed network devices such as the Berkeley motes, their parts are significantly less expensive and they can operate for substantially longer time periods.

5.2 Deployment: Combining Tools

Direct study of the sensor data may be useful to some researchers. For instance, the total number of firings or frequency of firings of particular sensors may be interest for specific inquiries (e.g. a sensor in a cabinet or drawer may offer clues about medication adherence or a light switch in the bedroom may offer clues about sleeping patterns). One of us, for example, is studying how the data may be used to help people learn about the use of their own environments (e.g. the kitchen) and to help them design new ones. Figure 3a shows one sequence of activations in a subject's home, where the arrows indicate sensors that fired after each other. This representation permits a researcher to study patterns of movement about the environment.

The sensor data collected, however, is most useful when the sensor readings can be correlated with self-report or observational data about what the person or people in an environment were actually doing at the time when objects in the environment were being manipulated. These activity labels are particularly important if the observation system is to be used to automatically collect data about behavior. With such labels, it then becomes possible to consider the creation of algorithms that recognize context automatically from the ubiquitous switch sensors using pattern classifiers customized to the individual using supervised learning (e.g. [4]).

We are exploring the use of two strategies to obtain the activity label data. The first is by using the context-aware experience sampling tool. In this approach, the environment sensors have been installed in homes of subjects who live alone. The subjects are given a PDA running the experience sampling software described in Sect. 4. As the state-change sensors record data about movement of objects, the subject uses experience sampling to record information about his or her activities. We have used a high sampling rate, where the subject is beeped once every 15 minutes for up to 16 days. At the beep, the subject receives the following series of questions. First the user is asked "What are you doing at the beep?". The subject selects the activity that best matches the one that he/she was doing at the time of the beep from a menu showing up to 33 activities. Next, we ask "For how long have you been doing this activity?" The subject can select from a list of four choices: less than 2 min., less than 5 min, less than 10 min., and more than 10 min. Then, the user is asked, "Were you doing another activity before the beep?". If the user responds positively, the user is presented with menu of 33 activities once again. For our current studies, we are using an adaptation of the activity categories used in the multi-national time-use study [27] supplemented with some activities from the compendium of physical activities [2].

The self-reported activities can be used to study the environment sensor data, or visa versa. Figure 3b shows all the environment sensor activations for one subject on a particular day at about the time a “cooking breakfast” activity was recorded by the context-aware experience sampling tool. Figure 3c shows the sensor activations for this same activity on a different day for the same subject.

In our work to date, we have found that subjects quickly acclimate to the presence of the environmental sensors. However, they have reported that the experience sampling software (sampling at roughly 15 minute intervals) becomes a significant burden after about 1-2 days. Interestingly, the interruption itself does not appear to cause the greatest annoyance (see Sect. 6. We are currently conducting tests to qualitatively measure ESM compliance and the benefits of context-aware ESM.

The second method to obtain activity labels we are exploring is direct observation. Fifty of the environment sensors have been deployed in a researcher's home with a wireless webcam that captures 30-100 frames per hour (based on a motion trigger) and saves the images to a nearby computer. Software can then correlate the timestamped imagery with the environment sensor readings. Figure 3d shows an image obtained from the webcam system. Such images can be annotated based upon the interests of the observing researchers. Alternatively, the environment sensor triggers can be used to find images that can be used for photographic analysis where a researchers asks a subject, “What were you doing here?”. Annotation of image data with other sensor data is useful for a variety of applications and analysis tasks [9,7]. In our system, the ESM, state-sensor,

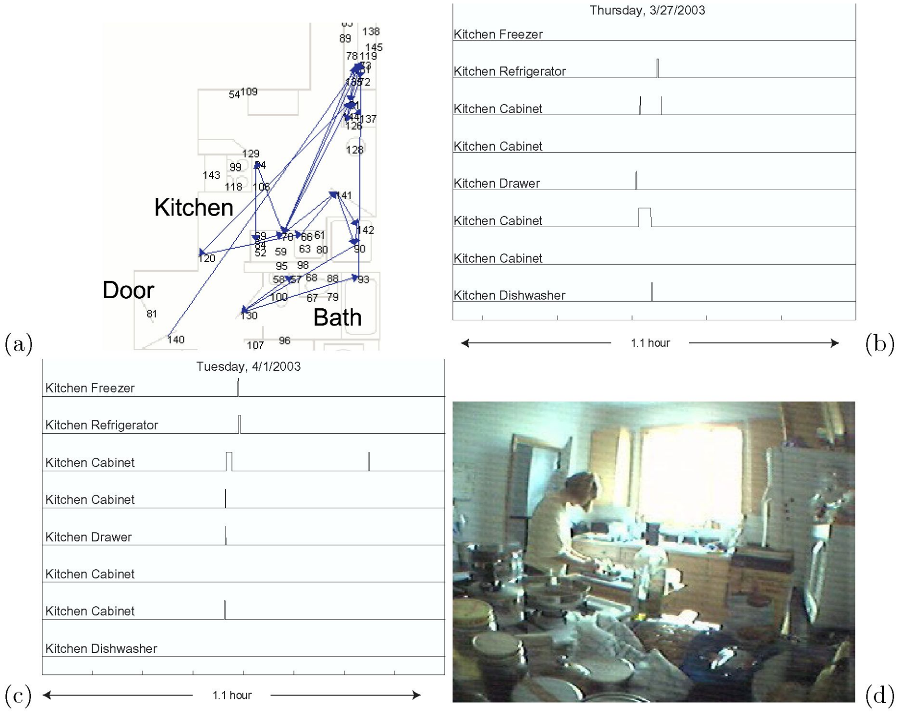


Fig. 3. (a) Data from the environment sensor system can be analyzed for clustering. This image shows the movements of one subject during a cooking event based upon sensor activation times. (b,c) The environment sensors are being used simultaneously with the context-aware experience sampling system to create datasets for pattern recognition research. These images show the environment sensors that fired around the time the subject reported a "cooking breakfast" event on two different days. (d) Time-lapse image capture is being correlated with the environment sensor data in some studies. Here a researcher works in her kitchen, which has been wired with environment sensors and a wireless camera.

and imagery data are correlated in time, which permits researchers to view only those portions of the data related to the event of interest.

5.3 Tool 3: Image-Based Experience Sampling

The third observational tool, which can be used in highly-instrumented environments, combines scene-based sensing with sampling techniques. ESM is less susceptible to subject recall errors than other self-report feedback elicitation methods, but the primary drawback to ESM is the interruption created by the sampling itself. ESM disrupts the user's activity, requiring the user to stop the current activity and answer questions. Therefore, we propose the use of context-sensitive image capture we call image-based experience sampling. Instead of disrupting the current activity with an alert, an audio-visual "image" is captured of the current activity at each sample time. The image can be a static picture or a small video clip of activity taken from one of potentially many recording devices installed in the environment.

This tool can sample on a fixed schedule, as the system that captured Fig. 3d is doing, or it can sample based upon sensor readings. We have implemented a context-aware prototype of this tool in our laboratory using a multi-camera computer vision person detection system. Samples are taken (i.e. images are captured) only when a person is identified as being in the room. Figure 4a shows one such picture that was captured when someone was sitting in the room. The sampling itself does not interrupt the activity or require any proactive action on the part of the person.

At some later time, a researcher can study the images. However, the full power of the technique is realized when not only the researcher but the user reviews the sampled imagery. The user was not interrupted at the sample time, but the rich contextual information provided by the image or video clip triggers the user's memory of the moment when the sample was taken [9]. Therefore, specific questions of interest can be asked of the user about that situation in time. A mobile interface allows the user to view the images and answer questions whenever it is convenient, such as during "idle time" waiting in line or riding transportation.

5.4 An Example Application

Our prototype shows off how this observational technique can be used to drive a new type of application: helping users make design decisions. Determining requirements for any design project involves identifying and ranking user needs and preferences. Image-based experience sampling can be used to assist someone who is interested in learning about his or her own preferences in a way that is more personalized and less disruptive than interviews, focus groups, and standard experience sampling.

Consider this scenario. Susan feels unhappy with her current kitchen and plans to remodel it sometime in the next six months. She has a limited budget and knows that she must prioritize the changes she would like to make. Such a scenario is common. The user knows that something in his or her life needs to change but is uncertain how to evaluate the relative importance of different options. A common method for helping users evaluate their preferences in such instances is to construct interfaces that prompt the user for information that is used to determine the combination of attributes that provides the most perceived value to the user [20]. Interfaces that use such an approach are typically prompting the user about preferences outside of the context of everyday activity (i.e. kitchen redesign software might ask Susan a series of questions about her kitchen). The best time to ask the user about preferences, however, would be in the midst of the actual activity being scrutinized. Further, most desirable would be if an interface could help the user build up awareness and understanding of the user's preferences over time. Combining image-based experience sampling and conjoint analysis, a technique for measuring preferences, a non-disruptive

(a) (b)

Fig. 4. In a prototype implementation of image-based experience sampling in our laboratory, images are captured when someone is in a room instrumented with cameras. At some later time, the user can view the captured images on a mobile computing device and rank preferences about the architectural space, as shown in (a) and (b). Conjoint analysis is then used to create an overall ranking of preferences, but those preferences are based upon samples taken in context of everyday life rather than using unaided retrospective recall.

but context-sensitive preference elicitation user interface mechanism can be developed.

Conjoint analysis is typically implemented via a written or online survey. A large set of features is preference ranked by asking users to compare smaller sets of features against each other. For example, Susan might be asked, "Choose a number from 1 (always) to 10 (never) indicating how often each of the following situations occur: kitchen counter too crowded, I feel tired, shoulders get sore, feel somewhat drab." If Susan answers a sufficient number of such questions, statistical analysis can be used to weight the relative importance of features and conditions.

Figure 4 illustrates the interface's operation. Using the cameras located in the environment, samples are randomly acquired whenever the user is in the space. Each day a few new images are acquired. The goal is to acquire samples of everyday activity so that the user can later refer back to remember a particular situation and comment on how the user was feeling about the environment.

At the user's convenience, subsets of images can be reviewed on a mobile computing device. Our model is that the user interacts with the interface in bursts of 30-90 seconds during idle moments throughout the day. The user can quickly scan an image or two and provide some preference information. This creates awareness and learning during short bursts of activity on a regular basis. Upon viewing a picture, the user will see the image or video clip, the time the clip was taken, and a question used to acquire information about how the person was feeling about the adequacy of the physical environment during the pictured activity.

Image-based experience sampling combines the power of three techniques: use of media for contextual recall, use of experience sampling, and use of conjoint analysis for preference rankings. In this example we implemented context-awareness (i.e. detecting if a person is around) using optical sensing. However, if imagery is captured continuously, the environment sensors could be used to identify specific events about which to later ask the user. For instance, the user could be shown images taken just before making a change to the lighting and asked to evaluate lighting preferences in the home.

6 Applications and Next Steps

As we deploy these tools and interview our subjects, we are learning how to improve them. For example, after several environmental sensor and ESM simultaneous deployments, we decided to add new functionality to the ESM software that permits a subject to fluidly switch between prompted data collection and self report. We found users develop mental models of how the experience sampling works that influences how they use it and that can contribute to feelings of frustration. For instance, subjects have expressed that they do not mind answering questions about new actions, but they get frustrated quickly when the ESM software is asking them questions about things they feel they have already "taught it." We are working to improve the context-aware experience sampling by giving the user more control (or the perception of more control) over the device's question-asking behavior. We are also working to add new functionality to each of the tools so they can be used more effectively in combination. For instance, the ESM software can now sample by taking pictures from the PDA directly. If worn in the front shirt pocket, the researcher can then obtain a continuous stream of pictures of what a person was facing. Because the system is not taking a picture of the user but instead of what the user sees, the system may be perceived as less invasive. The images, however, can improve a researcher's ability to interpret self-reported data or the data from the environmental switches.

These three tools are relatively new and yet we are rapidly expanding the number of uses we find for them – particularly the combination of the three techniques. Further, although image-based experience sampling is in use only as a laboratory prototype, the context-aware experience sampling tool and the environment sensing system have been deployed in multiple, non-laboratory settings for data collection. The tools are being used for the following ongoing work.

Mobile activity recognition algorithm development. The context-aware experience sampling tool is being used to collect datasets on physical activity for the development of algorithms that can detect various types of everyday activities automatically (e.g. walking, going to work, climbing stairs) using body-mounted accelerometers and a GPS plug-in. The results of some of our work on algorithms that automatically identify context that uses the tool will be rolled back into the tool itself, providing future researchers with more context-specific triggers.

Environment activity recognition algorithm development. The context-aware experience sampling tool in combination with the environmental sensors are being used to collect multiple datasets from *real homes* that can be used to study the development of new pattern recognition algorithms to detect everyday routines.

Interruptibility. We, like others [14], have found experience sampling to provide a valuable tool for studying detection of interruptibility, a key emerging problem for ubicomp acceptability. However, we are also using the context-aware capabilities of the context-aware experience sampling device to study the relationship between bio-monitored data and interruptible moments.

Kitchen design. We are using the environmental sensors with video image capture to study how these tools could be used to help people learn about their own behavior, particularly with respect to how that understanding might impact how people make future design decisions.

Other studies are also being considered, such as using the context-aware experience sampling device in a small workplace. Using a Bluetooth sensor the device could be programmed to detect moments of interaction between two people (within Bluetooth range). This particular context-aware trigger would permit the study of how casual interaction impacts work and the perception of productivity.

We have designed these tools to meet the needs of a set of ongoing and planned experiments in our laboratory, but we believe they are sufficiently robust to be of value to other researchers. Although the tools have only recently been deployed, we have been surprised at the number of new studies of people in naturalistic environments that they enable. Therefore, we have created an open source project to further develop the context-aware experience sampling tool [6], and the specifications for the hardware and software design of the environment sensors are available on request.

Finally, an apartment living laboratory is currently under construction in Cambridge, Massachusetts and will be completed by October, 2003. This lab will be fully instrumented with same sensor infrastructure discussed here. In particular, the environmental sensing system will be physically wired so that PDA-based sampling or image-based sampling protocols can be developed where particular actions (e.g. opening a cabinet, turning a faucet) can trigger sampling. The facility will be a shared scientific resource available to researchers. Studies will be possible using a combination of tools in stages: (1) first using the portable tools to study subjects in their own homes, then (2) studying the same subjects as they live temporarily in the living laboratory, and then (3) studying the same subjects as the move back into their own homes. We encourage researchers in the ubicomp community interested in using or helping to further develop these observational tools to contact us.

Acknowledgments

This work was supported in part by the NSF (#0112900), the House_*n* Research Consortium, and the AARP's Andrus Foundation. We would also like to thank the subjects who have graciously agreed to allow us to bring these sensing devices into their lives and homes.

References

1. G.D. Abowd. Personal communication, November 2002. Director of the Aware-Home Initiative, Georgia Insitute of Technology.
2. B.E. Ainsworth, W.L. Haskell, M.C. Whitt, M.L. Irwin, A.M. Swartz, S.J. Strath, W.L. O'Brien, Jr. Bassett, D.R., K.H. Schmitz, P.O. Emplaincourt, Jr. Jacobs, D.R., and A.S. Leon. Compendium of physical activities: An update of activity codes and MET intensities. *Medicine & Science in Sports & Exercise*, 32(9 Suppl):S498–504, 2000.
3. M. Alwan. Personal communication, November 2002. Assistant Professor, University of Virginia.
4. T. Barger, M. Alwan, S. Kell, B. Turner, S. Wood, and A. Naidu. Objective remote assessment of activities of daily living: Analysis of meal preparation patterns. Poster presentation, Medical Automation Research Center, University of Virginia Health System, 2002.
5. L.F. Barrett and D.J. Barrett. An introduction to computerized experience sampling in psychology. *Social Science Computer Review*, 19(2):175–185, 2001.
6. CAES. Context-Aware Experience Sampling Website, Access date: January 15 2003. http://caes.sourceforge.net.
7. P. Castro, M. Mani, S. Mathur, and R. Muntz. Managing context for internet video conferences: The multimedia internet recorder and archive. *Multimedia and Computer Networks*, 2000.
8. M. Coen. Design principles for intelligent environments. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*. AAAI Press, 1998.
9. J. Collier and M. Collier. *Visual Anthropology: Photography as a Research Method*. University of New Mexico Press, Albuquerque, rev. and expanded edition, 1986.
10. S. Consolvo, L. Arnstein, and B.R. Franza. User study techniques in the design and evaluation of a ubicomp environment. In G. Borriello and L.E. Holmquist, editors, *Proceedings of the Fourth International Conference on Ubiquitous Computing*, pages 73–90. Springer-Verlag, Berlin, 2002.
11. M. Csikszentmihalyi and R. Larson. Validity and reliability of the Experience-Sampling Method. *The Journal of Nervous and Mental Disease*, 175(9):526–36, 1987.
12. S. Hollar. *COTS Dust*. Ph.D. thesis, University of California, Berkeley, 2001.
13. J.M. Hudson, J. Christensen, W.A. Kellogg, and T. Erickson. 'I'd Be Overwhelmed, But It's Just One More Thing to Do:' Availability and Interruption in Research Management. In *Proceedings of the Confererence on Human Factors in Computing Systems*. ACM Press, 2002.
14. S.E. Hudson, J. Fogarty, C.G. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J.C. Lee, and J. Yang. Predicting human interruptability with sensors: A Wizard of Oz feasibility study. In *Proceedings of the Conference on Human Factors and Computing*. ACM Press, 2003.
15. S.S. Intille. Designing a Home of the Future. *IEEE Pervasive Computing*, April-June:80–86, 2002.
16. S.S. Intille, C. Kukla, and X. Ma. Eliciting user preferences using image-based experience sampling and reflection. In *Proceedings of the Conference on Computer Human Interface Short Abstracts*, pages 738–739. ACM Press, 2002.
17. S.S. Intille, J. Rondoni, C. Kukla, I. Anacona, and L. Bao. A context-aware experience sampling tool. In *Proceedings of the Conference on Human Factors and Computing Systems: Extended Abstracts*. ACM Press, 2003.

18. J.M. Kahn, R.H. Katz, and K.S.J. Pister. Mobile networking for Smart Dust. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 99)*. 1999.
19. O. Kasten and M. Langheinrich. First experiences with Bluetooth in the Smart-Its distributed sensor network. In *Workshop on Ubiquitous Computing and Communications, PACT*. 2001.
20. V. Mahajan and J. Wind. New product models: Practice, shortcomings and desired improvements. *The Journal of Product Innovation Management*, 9(2):128–139, 1992.
21. A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications*. 2002.
22. M. Mozer. The Neural Network House: an environment that adapts to its inhabitants. In *Proceedings of the AAAI Spring Symposium on Intelligent Environments*, Technical Report SS-98-02, pages 110–114. AAAI Press, Menlo Park, CA, 1998.
23. J.P. Robinson. The validity and reliability of diaries versus alternative time use measures. In F.T. Juster and F.P. Stafford, editors, *Time Goods and Well-Being*, pages 33–62. Ann Arbor, MI, 1999.
24. J. Rodin and E.J. Langer. Long-term effects of a control-relevant intervention with the institutionalized aged. *Journal of Personality and Social Psychology*, 35(12):897–902, 1977.
25. D.A. Schèon. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, New York, 1983.
26. A.A. Stone and S. Shiffman. Ecological momentary assessment (EMA) in behavioral medicine. *Annals of Behavioral Medicine*, 16(3):199–202, 1994.
27. S. Szalai. *The Use of Time. Daily Activities of Urban and Suburban Populations in Twelve Countries*. Mouton, The Hague, 1973.

Very Low-Cost Sensing and Communication Using Bidirectional LEDs

Paul Dietz, William Yerazunis, and Darren Leigh

Mitsubishi Electric Research Laboratories
201 Broadway, Cambridge, Massachusetts 02139 USA
{dietz,wsy,leigh}@merl.com

Abstract. A novel microprocessor interface circuit is described which can alternately emit and detect light using only an LED, two digital I/O pins and a single current limiting resistor. This technique is first applied to create a smart illumination system that uses a single LED as both light source and sensor. We then present several devices that use an LED as a generic wireless serial data port. An important implication of this work is that every LED connected to a microprocessor can be thought of as a wireless two-way communication port. We present this technology as a solution to the "last centimeter problem", because it permits disparate devices to communicate with each other simply and cheaply with minimal design modification.

1 Introduction

Light Emitting Diodes, or LEDs, are one of the most common types of interface components. Their diverse applications include numeric displays, flashlights, liquid crystal backlights, vehicle brake lights, traffic signals and the ubiquitous power-on indicator light.

Because LEDs are so commonly used as light emitters it is easy to forget that they are fundamentally photodiodes, and as such, are light detectors as well. Although LEDs are not optimized for light detection, they are very effective at it. This interchangeability between solid-state light emission and detection was widely publicized in the 1970's by Forrest W. Mims [1][2], but has been largely forgotten by LED users.

1.1 Ambient Illumination Sensing with LEDs

Recently, we have been investigating improvements for infrared remote controls of the type commonly used with consumer audio/video equipment. An area of immediate interest was the pushbutton illumination used on many remote controls. To activate the backlight, you must press a button that is nearly impossible to locate in the dark! We resolved to rectify this situation.

Our first solution was to use a capacitive proximity sensor (similar to the one described in [4]) to activate the remote control backlight during active handling. Unfortunately, turning on the backlight *every* time the remote is handled

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 175–191, 2003.

substantially decreases battery life, not only because the user often holds onto the remote continuously but also because the remote is sometimes used under good lighting conditions when the backlight is not needed. While a mode switch could be added, this would be little better than the original situation.

The obvious step was to add a light sensor to turn on the backlight only when needed. CdS photocells are inexpensive, but providing an optical path to the cell would add significant cost and complexity to the mechanical design. Recalling the photosensitive nature of LEDs, we decided to investigate using the backlight LED itself as the light detector. We developed a simple circuit for this purpose that requires one additional microcontroller I/O pin, but no other additional components compared to a traditional LED driver.

The success of the simple LED emitter/receiver circuit inspired us to consider other applications. For example, by quickly switching between the forward-biased (light-emitting) and back-biased (light-sensing) modes, it is possible to build an LED-based light source that appears to be constantly on, but is in fact periodically measuring the ambient lighting level and using this information to automatically adjust the brightness level of the LED. Our demonstration device, shown in Figure 6, has a capacitance sensor to determine that the device is being manipulated and an LED sensor/emitter to provide the backlight function.

1.2 LEDComm: Bidirectional LED Communication

While the measurement of ambient light levels has many applications, a more intriguing use of this technology is to transmit data back and forth between LEDs pointed at each other. We call this *LEDComm*. We have developed simple prototypes that allow two-way serial data communication between LEDs over a distance of several centimeters.

One possible application of LEDComm is to replace Radio Frequency Identification (RFID) systems (e.g. [5]) for payment authorization and access control. To test this concept, we have created an inexpensive keychain-size device called an *iDropper* that can receive, store, and transmit data. Unlike RFID systems, iDroppers support true peer-to-peer communication, allowing new functionality such as directly transferring authority between devices without need of a special reprogramming station.

The implications of LED-based data communication are significant. *Every LED connected to a microprocessor can be thought of as a wireless communication port.* Compared with other short-range wireless technologies such as IrDA [7] and Bluetooth [8], LEDComm has a far more limited range and a much slower data rate. But LEDComm can be implemented at a fraction of the cost, and in many cases, may even be free. This is because LEDComm is essentially a software interface technique using existing hardware with minimal modification.

LEDComm allows us to implement communication functions in places where traditional techniques are too expensive. The power light on many consumer appliances can now become a maintenance port for reading service information or uploading new firmware. Cell phones can transfer contact information to other phones by holding their displays next to each other. For automobiles, the standard expensive service connector can be bypassed, and all data transferred through the "Check Engine" light. (An automobile owner could even use an iDropper to capture the car's fault log and transmit it to the service center before a service appointment, insuring that the proper tools and spare parts will be immediately available when the vehicle is brought in.) There are many possible applications.

In the following sections, we describe the basic bidirectional LED microprocessor interface circuit and its use in the smart backlight. We then give a full description of LEDComm, iDroppers and various applications.

2 The Bidirectional LED Interface

Light emitting diodes emit light in a fairly narrow frequency band when a small current is applied in the correct direction. Because the current-voltage characteristic is exponential, it is difficult to control a voltage applied directly across an LED accurately enough to attain a desired current; some means must be used to limit the current. In discrete systems, this is typically done by placing a resistor in series as shown in Figure 1. Since most microprocessor I/O pins can sink more current than they can source, the configuration shown in the figure is the most common way of driving an LED from a microprocessor.

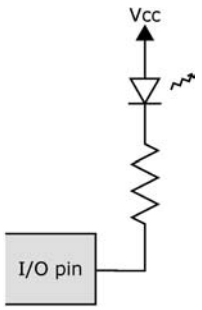


Fig. 1. Schematic of a typical LED driver.

The LED is a photodiode that is sensitive to light at and above the wavelength at which it emits (barring any filtering effects of a colored plastic package). Under reverse bias conditions, a simple model for the LED is a capacitor in parallel with a current source which models the optically induced photocurrent. (see Figure 2, [3]). It is this photocurrent that we would like to measure.

An inexpensive way to make a photodetector out of an LED is to tie the anode to ground and connect the cathode to a CMOS I/O pin driven high. This reverse biases the diode, and charges the capacitance. Next switch the I/O pin to input mode, which allows the photocurrent to discharge the capacitance down to the digital input threshold. By timing how long this takes, we get a

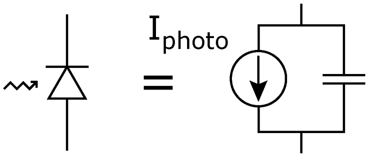


Fig. 2. Reverse-biasing an LED for photosensing.

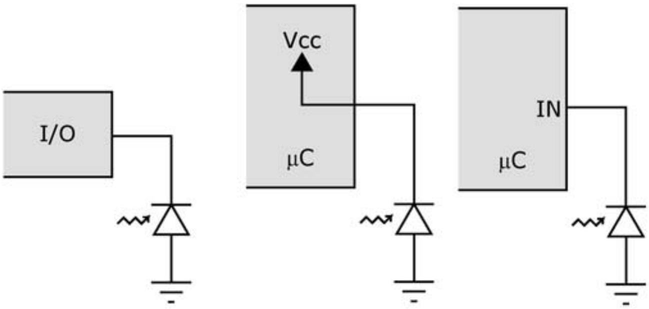


Fig. 3. LED used as a photosensor.

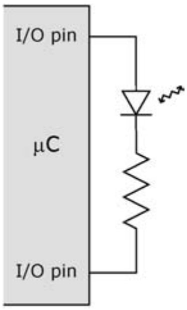


Fig. 4. Schematic of a bidirectional LED interface.

measurement of the photocurrent and thus the amount of incident light. This sequence is shown in Figure 3.

The circuits of Figures 1 and 2 can be combined to create a general bidirectional microprocessor interface to an LED as shown in Figure 4. This is identical to the circuit of Figure 1, except that now the resistor/LED combination is placed between two I/O pins.

Figure 5 shows how the pins are driven for the two modes. Figure 5a shows the "Emitting" mode where current is driven in the forward direction, lighting the LED. Figure 5b shows "Reverse Bias" mode, which charges the capacitance and prepares the system for measurement. The actual measurement is made in "Discharge" mode shown in Figure 5c. Since the current flowing into a CMOS input is extremely small, the low value current limiting resistor has little impact on the voltage seen at the input pin. As before, we simply time how long it takes for the photocurrent to discharge the capacitance to the pin's digital input threshold. The result is a simple circuit that can switch between emitting and receiving light.

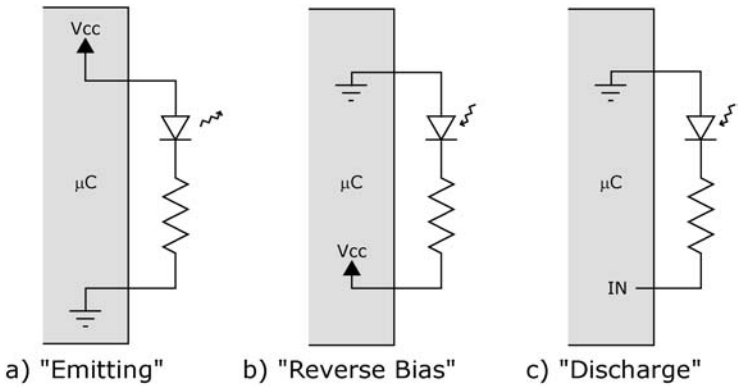


Fig. 5. Emitting and sensing light with an LED.

Because the circuit changes required to provide this bidirectional communication feature consist of only one additional I/O pin and printed circuit board trace (which can be provided at design time for zero additional hardware cost) we claim that adding this hardware feature to a device is essentially free. Of course, software and CPU runtime are also necessary to make this work.

Compare this to the cost of adding IrDA [7] (about $7) or Bluetooth [8] (more than $10) to a product. Using even a simple mechanical connector can cost several dollars because of the required level-shifting and electrostatic discharge (ESD) protection circuitry. Using an existing LED for communication can also save manufacturing costs because expensive plastic molds for the housing need not be altered to accommodate a dedicated infrared transceiver, antenna or physical connector.

3 The Smart Backlight

The *smart backlight* is one application of the bidirectional LED circuit. As noted previously, the idea of the smart remote control backlight is to turn on the backlighting *before* the user has to press a button. Also, to conserve power, we wish to turn on this backlight only when it is actually dark enough to need it.

To demonstrate this function we created the prototype shown in Figure 6, with the complete schematic shown in Figure 7. This circuit uses a capacitive proximity detector to determine handling state. Although the basic capacitance measurement circuit is identical to that used in the buffer phone [4], we process the data to look for active handling (changes in capacitance) rather than simple

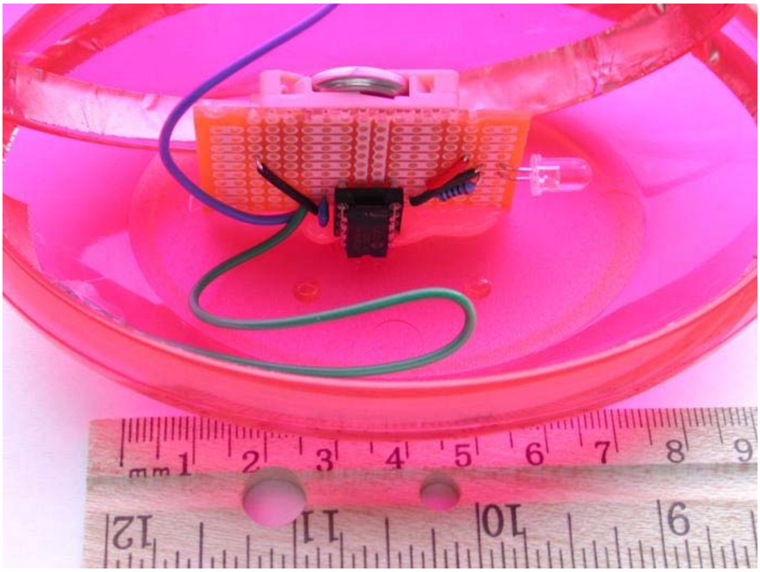


Fig. 6. The smart backlight prototype.

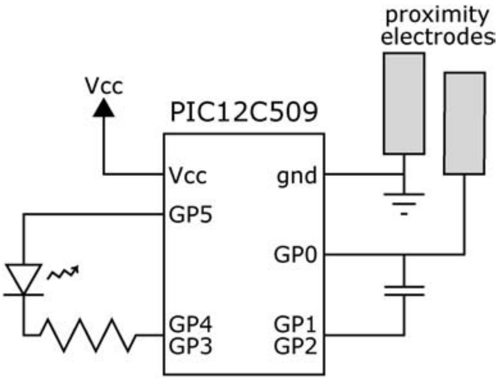


Fig. 7. Schematic for the automatic backlight.

presence (increased capacitance). Many users will continue to hold a remote even when they are not actively using it, so the detection of active handling is critical for extending battery life. Of course, as soon as the user wishes to actively use the remote again, any significant motion turns the light back on.

The smart backlight functions as follows: periodically, the microprocessor wakes from sleep and measures the capacitance. If no active handling is detected, the processor goes back to sleep. Otherwise, a light measurement is made with the LED. If the room is dark, it turns on the backlight for at least two seconds. While the backlight is on, it continues to check for active handling. Each time

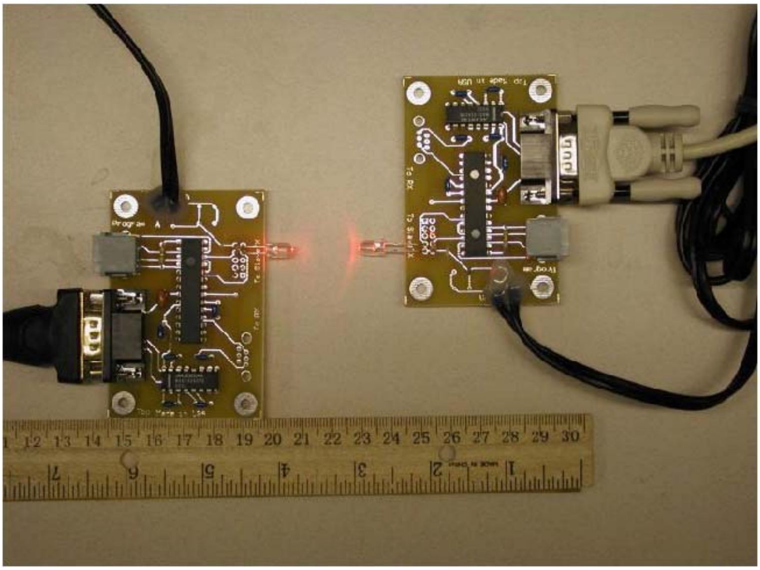


Fig. 8. Bidirectional communication with LEDs.

handling is detected, the backlight timer is reset to stay on for another two seconds.

Since remote controls already contain low-end microprocessors, adding this functionality costs very little. The proximity electrodes can be part of the printed circuit board, eliminating the need for special tooling. If there are spare I/O pins available, the only additional component is a single, inexpensive capacitor for the capacitance sensor.

One might wonder if the constantly running proximity detector adversely impacts battery life. In fact, the circuit draws microwatts of power; the prototype ran continuously for 6 months on a single type CR2032 coin-cell "watch" battery. Remote controls typically use AAA or AA batteries with a storage capacity an order of magnitude higher than the coin-cell, so the power draw would be insignificant compared to the batteries' self-discharge characteristics.

4 Bidirectional Communication Protocols

In our initial experimentation with the smart backlight, we often used LED-based flashlights to test the light detecting circuit. This suggested to us that LED-to-LED communication was feasible. We constructed a simple test setup using two identical, generic PIC microcontroller boards with RS-232 interfaces as shown in Figure 8.

These test boards use a simple protocol for data transfer which allows two unsynchronized devices to phase-lock to each other and exchange pulse-width-modulated data bidirectionally. A basic explanation of the protocol is that the

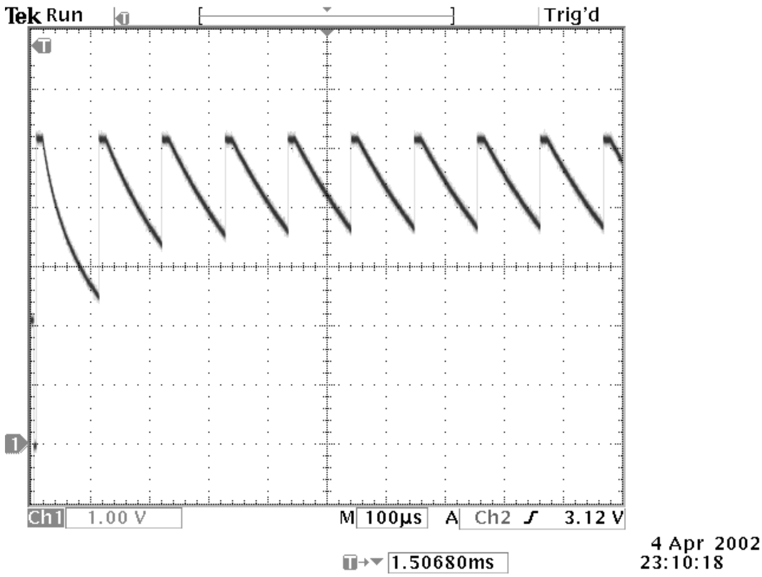


Fig. 9. A series of light measurements under normal room illumination.

two devices take turns flashing their LEDs at each other. A short flash indicates a 0 or SPACE state, and a long flash indicates a 1 or MARK state.

The protocol starts out on powerup with the device performing an idling cycle, transmitting a 1 millisecond light pulse followed by a 4 millisecond receive period. During the receive period, the device executes 40 light measurements, each one taking 100 microseconds. These light measurements provide only one bit of resolution, i.e. whether the incoming light flux is above or below the digital I/O pin's threshold (nominally about 1.5 volts). With only normal room light incident upon the LED there is insufficient photocurrent to discharge the capacitance below the threshold during the 100 microsecond receive period.

The oscilloscope trace in Figure 9 shows the voltage at the LED cathode during several light measurements with normal illumination. The vertical scale is 1 volt/division and the horizontal is 100 microseconds/division. The capacitance is initially charged to about 5 volts and then allowed to discharge. Notice that the voltage never drops below the threshold and so the microcontroller will always read the pin as a 1.

Figure 10 is an oscilloscope trace of the same setup, but with the LED being illuminated by another LED. The capacitance discharges completely during the measurement period, bringing the I/O pin voltage well below threshold and causing the pin to read as a 0. The idling cycle continues until at least two measurement times in succession indicate "light seen". At this point, the device assumes an incoming pulse of light from a similar device has been detected, and

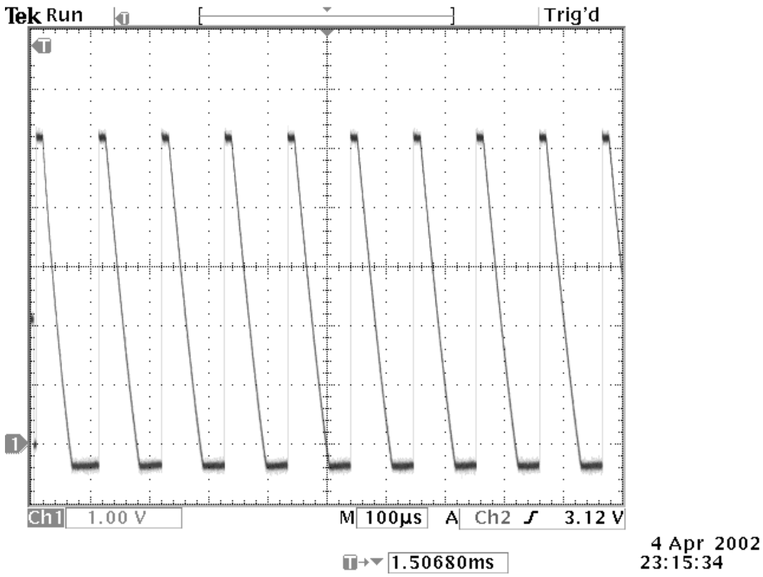


Fig. 10. A series of light measurements under LED illumination.

shifts from the idling loop of 1 millisecond ON then 4 milliseconds OFF to a slightly faster synchronizing loop, described next.

During the synchronizing loop, the transmitted light pulse is still 1 millisecond ON, but followed by a variable number of 100 microsecond light measurements. When in the synchronizing loop, the microcontroller will terminate the measurement set after either 40 are performed, or when the trailing edge of a light pulse is detected. A trailing edge is considered to be found when a pair of back-to-back measurements both indicate "light seen" followed by ten measurements without "light seen".

The execution pattern inside the synchronize loop is therefore composed of one device's LED on for 1 millisecond, then a 1 millisecond period with both LEDs off, followed by the other device's LED on for 1 millisecond, and finally both LEDs off for 1 millisecond. Even if the devices have clock frequency errors of up to 25%, they will still be able to synchronize. The nominal synchronize loop pulse rate is 250 Hz, with a 25% duty cycle. Figure 11 shows an oscilloscope trace of two devices in the synchronize loop, firing pulses of light at each other. Note that their clocks are completely independent and that all synchronization is occurring via the LEDs and the base protocol.

During communication, data bits are transmitted in asynchronous form. A 1 millisecond light pulse indicates a MARK and a 0.5 millisecond light pulse indicates a SPACE. The system normally idles with MARK bits being transmitted (the data transfer loop is the same software as the synchronize loop). During data transmission, the format starts with a single SPACE as a start bit, followed

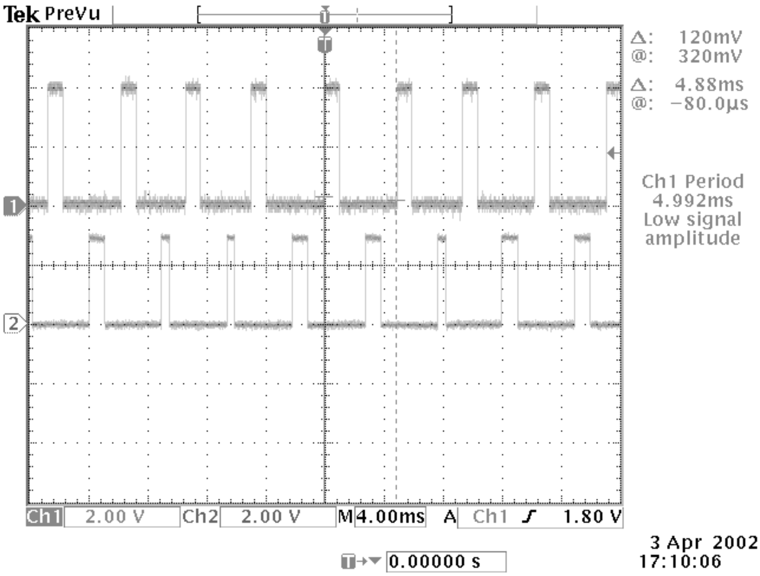


Fig. 11. Oscilloscope trace of two devices in synchronized operation.

by eight bits of data, followed by one MARK as a stop bit. This is similar to the common 8-N-1 RS-232 format. The top trace of Figure 11 shows the data pulse train of a device that is idling, sending all MARK pulses. The bottom trace shows a device sending both narrow SPACE and wide MARK pulses.

To decode the light pulses, the receiving device keeps a count of "light seen" measurements for each execution of the synchronize loop. If seven or fewer light-seen measurements are tallied, a SPACE is recorded; if eight or more are seen, a MARK is recorded. The usual asynchronous deframing (dropping the leading SPACE start bit and the trailing MARK stop bit) is performed. The resulting 8-bit data word is then available to the application-level program. A simple higher-level protocol allows for error detection and correction.

The LEDComm test setup works very well. The underlying protocol transmits data at a rate of approximately 250 bits/sec in each direction. The microprocessors buffer the data and connect to a host at 38400 bps. Data transfer is robust up to a range of approximately three centimeters. Because the LEDs we used have a fairly narrow beam angle, they permit a pointing error of only about 20 degrees.

Unlike many other protocols, this system is highly resistant to clock speed errors. Not all of our devices have precise crystal oscillators; some use the inaccurate, internal RC oscillator of the PIC microcontrollers. Even with errors in clock speed of up to 25%, communication is not disrupted. In contrast, errors over 5% in an RS-232 data link will often cause problems.

The cheap, unstable oscillators internal to the PIC devices are actually advantageous in this application. Even if two devices are powered up at the same time in the exact same phase relationship, they will quickly drift out of synchronization enough that a trailing edge will be detected and the devices will synchronize into alternating flashes. In our devices, this usually happens in under 50 milliseconds. If two LEDComm devices were to both have highly stable timebases (or if both devices derived their clocks from the same source), it would be necessary to insert a jitter source (perhaps based on a hash of the device serial number) into the idle loop to assure that the two pulse trains would drift out of phase enough for a pulse trailing edge to be detected.

An additional feature of this base protocol over a balanced pulse protocol, such as Manchester coding, is that the LED gives a visible indication of idle state vs. synchronized state vs. the data-transfer state. The perceived light brightens when ready to transfer data (due to the faster pulse repetition rate) and darkens during data transfer (due to the short 0.5 millisecond SPACE pulses versus the no-data 1 millisecond MARK pulses).

5 iDroppers

To act as tangible, portable repositories of information or authorization, we have designed and constructed a device that we call an *iDropper* (for Information Dropper). Like an eyedropper, an iDropper can suck up a small amount of information, hold it, and then expel the information on demand. Unlike an eyedropper, the iDropper can repeatedly expel the same information nondestructively.

The iDropper is meant to be used in situations where the user wishes to transfer data to or from devices that, for economic or practical reasons, do not have a viable user interface. This may be because the data transfer happens too infrequently to justify adding a display and keypad to the device, i.e. for diagnostic and initial setup information. An iDropper can be used to shuttle the data between the device and another which does have a user interface.

The tangible information appliance aspect of the iDropper is similar in effect to mediaBlocks [9]. The major difference is that mediaBlocks do not hold any information; they act as tokens for information that is passed along a network. The iDropper itself does hold information and can itself be used as part of the network.

The iDropper hardware is composed of a tiny printed circuit board, a single pushbutton switch (the sole user input), a Microchip PIC16LF628 microcontroller, an LED (which performs data input, data output, and user output), a 3 volt lithium "coin-cell" battery, a capacitor, and two resistors. There are an additional five solder pads so that extra components can be added for experimentation purposes. The entire assembly is smaller and cheaper than most car-alarm keychain remote controls and contains fewer components. A mass produced version should cost less than a dollar more than a similar LED keychain flashlight.

The prototype iDroppers are also equipped with an in-circuit programming connector which allows us to download code into the microcontroller and to change the personality of the device. We have also devised a small adapter board to convert this connector to Microchip's standard RJ-11 in-circuit debugging module. A pair of iDroppers is shown in Figure 12. The lower one has the adapter board attached. The large plastic part visible on the bottom of the lower iDropper is the battery holder – the largest component of the device. We have left over a centimeter of empty printed circuit board material at the back end of the iDropper so that a hole can be drilled and it can be attached to a keychain.

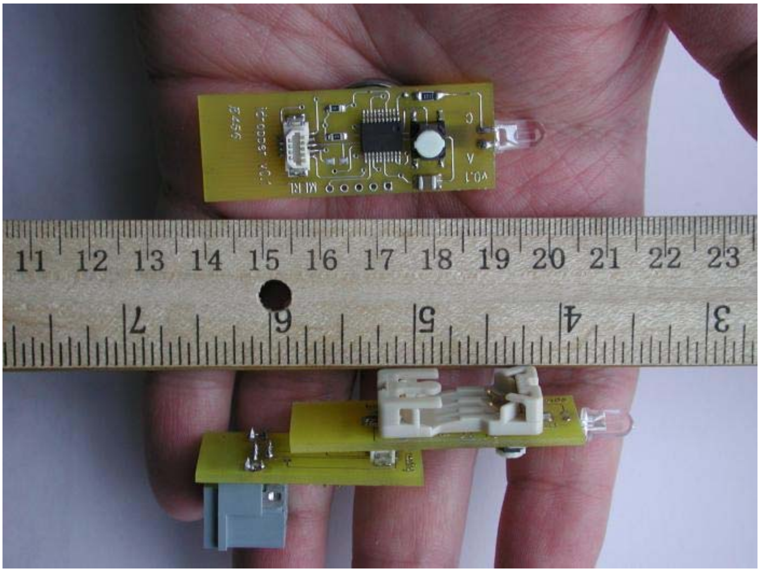


Fig. 12. A pair of iDroppers, one with programming adapter.

The default iDropper personality program is that of an information eyedropper. To suck information into the iDropper, the user presses the button twice and holds it in; the iDropper will then receive any data stream presented to it and store it internally. Distinctive flash patterns indicate when the mode has been entered and when recording has finished. Releasing the button early will abort the process. To squeeze information out of the iDropper, the user presses and holds the button; the data is then emitted repeatedly, about once a second. This mode appears to the eye like a simple flashlight. This is by design: an iDropper is perfectly useful as a small keychain flashlight.

The lithium battery employed in the device will allow over ten hours of continuous use. When an iDropper is not transmitting or receiving data, the PIC microcontroller goes into sleep mode. This lowers the power requirement for the entire system to below the leakage current of the battery, giving a shelf life of several years.

6 iDroppers as Intelligent Keys

One of our goals for the iDropper is to use it as an intelligent, programmable key. Although many other technologies are used in intelligent keys (RF and RFID, card-keys, etc.), LEDComm has some distinct advantages. First, it requires no physical contact so there is no mechanical wear unlike in some card-key systems. Second, unlike RF systems, it is directional and short range so the user has complete control over what is being unlocked. This allows a single key to be used for many different locks without the possibility of unlocking the wrong one just because it is nearby. Third, LEDComm is fundamentally bidirectional allowing the use of challenge/response and encryption protocols which can make the key very difficult to copy or spoof. Fourth, the visible nature of the LED allows for some user interface. (At the very least, the user can easily tell whether the device is operating or if the battery is dead.) Fifth, LEDComm readers are much easier and less expensive to implement than keycard or RFID readers. This could be important in situations where the number of locks is on the same order as the number of keys.

The sixth, and perhaps most interesting, advantage is that LEDComm is capable of peer-to-peer communication. Any LEDComm device can pass information or authorization to another LEDComm device (assuming the application software allows it). In this case, an iDropper with the standard "suck/squeeze" personality program can learn the unlock code, and pass that to yet more iDroppers. This ability to delegate authority is completely unique and not a capability of smart cards or RFID tags.

To demonstrate this use of the iDropper as an intelligent key, we added a reed relay and an external power supply to one device, and wired it into the security system that locks and unlocks our site's front door. The iDropper's LED was aimed out through the glass windows of the lobby. The iDropper personality program was altered to do nothing until it received the proper (secret) command, and when it received that command, to activate the relay to unlock the door for five seconds. Figure 13 shows the test setup.

We then programmed one iDropper with the correct code, and (as expected) used it to unlock the door. Taking advantage of the LEDComm peer-to-peer ability, we passed the unlock code to several other iDroppers which were also used to unlock the door.

7 iDroppers, Authentication and Security

In some applications, the peer-to-peer ability to transfer information or authorization is desirable. In other applications, such as financial and other secure transactions, authentication is as important as transfer, and the uncontrolled

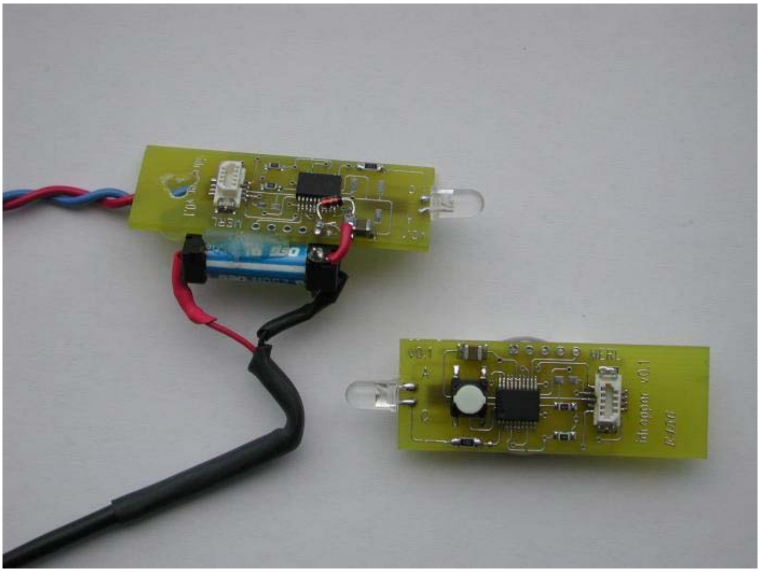


Fig. 13. iDroppers being used as a door lock (left) and key (right).

passing of authority must be prevented. An unfortunate side effect of the programmable nature of the iDropper is that there is no guarantee that another device will respect any "do not forward" data tags that may be inserted by an application. Non-transferable authorization and unforgeable proof-of-identity are difficult problems with many subtleties. A solution for even a highly constrained scenario would involve hardware, software and cryptographic techniques beyond the scope of this paper.

However, simple cryptography is quite possible and can be used to keep iDropper transactions secure from eavesdropping and spoofing. The microcontroller used has sufficient power to implement common symmetric cryptographic algorithms. These require the sender and recipient to share a secret key so communication between any two devices must be configured in advance. The iDropper has enough memory to hold many symmetric encryption keys and can therefore be set up to talk to a number of other devices.

Zero-knowledge proofs and public-key (or asymmetric) cryptography [6] would enable an iDropper to securely prove its identity and communicate with any device that had access to published information; no shared secrets would be necessary. Unfortunately, all published algorithms for these require extensive calculations and data communication which, although available on modern fast workstations, are not possible in the extremely limited computational environment of an iDropper. The small lithium battery contains only enough energy to run the 4 MHz processor for a few hours, so a calculation that a workstation could complete in seconds would consume an iDropper's entire battery life.

We are currently investigating algorithms for zero-knowledge proofs that require only the limited processing we have available.

One side effect of the limited data rate and battery lifetime of the iDropper is that the total amount of data it can communicate in that lifetime (about 10^6 bytes) could be held in an inexpensive non-volatile memory. This would allow the use of one-time pad encryption, either for extreme security or for use with a very limited microprocessor.

8 Every LED is a Communications Port

Although almost every electronic device made today contains a microcontroller and (theoretically) has sufficient capability to communicate with similar devices, the cost of including a communication link often precludes two devices that are sitting side-by-side from talking to each other. This is the "last centimeter problem".

With LEDComm technology, every LED becomes a potential communication path. This has broad implications because LEDs are widely used as power-on indicators in microcontroller-based devices. The indicator is usually not wired directly to the power supply, but is connected through the microcontroller so that a minimal user interface (some blinking) is available. With the proper modifications, the indicator can be used to communicate with an iDropper or other LEDComm-enabled device.

The ability to cheaply and easily transfer data between a device with a user interface and one without will permit designers to add more capability to inexpensive products. Small, portable products can be carried to the user interface machine for interaction there, while larger ones may require that the user carry data back and forth with an iDropper-like device. Here are some applications of LEDComm-enabled devices that we have been considering:

1. The power indicator LED of a modern CRT monitor is connected to its CPU so that it can blink to indicate a low-power "standby" state. Newer models are equipped with USB, both to control monitor settings and to provide easy access for mice and keyboards. Adding LEDComm can provide a complete data path from the power LED to the host computer, allowing an iDropper or similar device to be used as a key. This could be used instead of or in addition to a password to log in to the computer, or could be used as a cryptographic authentication device for electronic commerce. A similar technique could be used with keyboard indicator lights.
2. A homeowner could copy the full diagnostic state of his or her malfunctioning washing machine by iDroppering the data from the power-on LED and carrying it to their PC for upload to a service site on the web. No special display or connector on the washer is needed, nor is it necessary to run a data cable to the computer.
3. Exchange of telephone numbers and electronic business cards with a new acquaintance could be performed by holding two mobile phones together, display to display, while the backlight LEDs exchange the relevant data.

4. A programmable electronic doorbell might need to have its tune changed seasonally. This could easily be accomplished by composing or downloading a new tune on a computer and transferring that to the doorbell with an iDropper. There is no need to remove and carry the doorbell itself (with the attendant wiring difficulties) or to implement an expensive wireless data link.
5. LEDComm could be used for mobile phone-based electronic payment. A purchaser could use the user interface and wireless data connection of the telephone to set up an electronic payment transaction with their bank. They could then point the phone's LEDComm-enabled power LED at the vending machine of interest, completing the transaction. The LED's directionality and short range are an advantage here because they allow the user to specifically and naturally indicate for which machine the payment is intended.
6. Inexpensive toys using LEDComm could communicate with each other to synchronize their actions or provide emergent behavior among a group of related toys. They could also communicate with the family computer to interact with programs running there or to download new functionality.

9 Notes and Future Work

One caveat that we should mention is that the LEDComm technique exploits parameters of a microcontroller's CMOS digital I/O pins which are typically not well characterized. Noise, threshold drift and input leakage current will all affect performance. Of these, only input leakage current appears regularly on device datasheets, and even there the information is of dubious value. In many cases, leakage currents are far too small to be accurately measured by production line test equipment so manufacturers will specify the value as the smallest current the tester can reliably measure. This can be orders of magnitude larger than the actual leakage current.

Because the leakage current in a reverse biased p-n junction doubles approximately every 11 degrees Celsius [10] it is possible that, for some semiconductor processes, the LEDComm technique may have problems when operating at high temperatures. Software can compensate for this by using an adaptive threshold, but if the leakage becomes too great, the technique will fail. Until manufacturers fully specify I/O pin characteristics, it will be impossible to guarantee the reliability of LEDComm under all conditions.

The range of communication for an LEDComm device is currently quite short – a few centimeters at best. The data rate is also fixed at 250 bits/second in each direction. These two values (range and data rate) are inversely related: altering the base protocol to use longer integration times can yield a longer range, while increasing the data rate decreases the total integrated light captured by the LED, which lowers the signal-to-noise ratio and so limits the maximum distance between the LEDs.

We are currently perfecting improvements to the LEDComm hardware and software to permit operation at substantially longer ranges (greater than one meter) with somewhat slower data rates, as well as operation at over 1000 bits/second at the existing maximum range. The operating conditions can be detected by the system so that the data rate can be automatically raised or lowered as conditions permit.

LEDComm-enabled devices will have to become widespread to be useful. This will require a standardization process for the several layers of communication protocol, optical characteristics, etc.

References

1. Mims, Forrest M., III, *Siliconnections: Coming of Age in the Electronic Era*, McGraw-Hill, New York, NY, 1986.
2. Mims, Forrest M., III, *LED Circuits and Projects*, Howard W. Sams and Co., Inc., New York, NY, pp. 60-61, 76-77, 122-123.
3. Graeme, Jerald, *Photodiode Amplifiers: Op-amp Solutions*, McGraw-Hill, 1996, pp. 4-7.
4. Dietz, P. and Yerazunis, W., *Real-Time Audio Buffering for Telephone Applications*, in Proceedings of UIST 2001, Orlando, FL, Nov. 11-14, 2001, pp. 193-4.
5. Mobil Speedpass is an RFID payment system. Information is available at `http://speedpass.com`
6. Schneier, B., *Applied Cryptography*, second edition, John Wiley & Sons, New York, NY, 1996, pp. 101-111.
7. Infrared Data Associates (IrDA) information and specifications can be obtained at the association's web site: `http://www.irda.org`
8. Bluetooth information and specifications can be obtained from the Bluetooth SIG, Inc. website: `http://www.bluetooth.org`
9. Ullmer, B., Ishii, H. and Glas, D. *mediaBlocks: Physical Containers, Transports, and Controls for Online Media*, in Computer Graphics Proceedings (SIGGRAPH '98), July 19-24, 1998.
10. Johns, D. and Martin, K. *Analog Integrated Circuit Design*, John Wiley & Sons, New York, NY, 1997, p. 42

SPECs: Another Approach to Human Context and Activity Sensing Research, Using Tiny Peer-to-Peer Wireless Computers

Mik Lamming[1] and Denis Bohm[2]

[1] HP Labs, Palo Alto, CA
mik@hp.com
[2] Firefly Design, Los Altos, CA
denis@fireflydesign.com

Abstract. A small battery powered peer-to-peer proximity sensing platform that can be attached to people, places and things can be a valuable tool to conduct research in human activity sensing. Such a platform senses the subjects' presence and activities in a wide variety of contexts, for example home, car, work, or shopping. It eliminates the need for deployment and maintenance of prohibitively expensive infrastructure. The goal is to sense the activities of one individual at large in their world, rather than the activities of a group in a well-instrumented laboratory setting. Preliminary results with a real-world application are described.

1 Introduction

How should researchers explore pervasive applications if they can't afford to deploy pervasively the experimental infrastructure they need? This paper describes a platform we have built called SPEC, to support our research in human context and activity sensing.

1.1 Lessons from Earlier Work

In the early 1990s Xerox EuroPARC researchers started investigating how applications could benefit from knowledge of the user's activities [1] using Active Badges [2]. PARC, EuroPARC's sister lab, developed a more capable platform [3] to explore the systems issues in more depth and demonstrate the fundamental capabilities of what Weiser [4] coined "ubiquitous computing" technology. At EuroPARC a number of novel activity-sensing demonstrators were built, including Pepys [5] an automatic personal biographer, Forget-me-not [6] a personal memory aid; and a reminding system [7].

Although these groundbreaking systems showed considerable promise, conducting trials beyond the confines of the laboratory proved difficult and expensive for reasons we will detail later. Services that could be tested at work were not available in other contexts outside work. Many memory problems like, "What was the name of that

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 192–199, 2003.

person I met on the airplane last week?" involved accessing information from, in, or between non-work contexts. So Forget-me-not for example – a memory aid that could only be used at work, and then only to recall work events was frustratingly difficult to evaluate.

Pepys and Forget-me-not provided continuous service through a series of wireless base stations connected to a central server. Having to connect each base station to a server, albeit via a network, made deployment extremely time consuming and expensive. Connecting to distant locations, like homes, or mobile locations, such as our cars, incurred significant telecommunication service costs, and required expert knowledge of network and security systems.

Active badges achieved a uniquely useful balance between sensing-range, battery life, and wearability, capable of providing hassle-free service running into many months without human intervention. Larger, more capable devices are nevertheless less convenient to wear all the time, and need frequent recharging, and every recharge raises the possibility of the device being forgotten and left behind. At the other end of the scale, RFIDs are easy to incorporate into clothing, but the readers are relatively large, and have a short range.

We found our fellow researchers, in large part, quite willing to be subjects for experiments that had the clear potential to invade their privacy. However there were concerns voiced about accidental, unthinking, or perhaps insensitive uses of the accumulated data, which was all stored in a single centralized database. In larger installations where there were lower levels of trust some people simply opted-out, or quite reasonably wanted to see what they were revealing about themselves *before* their data appeared in the central repository. It became clear that this would be an issue for larger scale field trials.

1.2 Proximity Is Often Enough

Many location sensitive systems detect absolute position to some level of granularity using GPS, or cell-ID. To detect in real time who, or what, is nearby they update and then query a central database. If either sensing device is out of range of the locating or communications infrastructure (deep in a building for GPS, or out of the service area for cell phone-based location system), they won't be able to determine that they are co-located. Noticing that two sensors are co-located if they are both in a moving vehicle can be complex – especially if each updates the central database relatively infrequently. But for systems like Pepys it was sufficient to collect in real time only a unique identifier for the nearby objects, for later resolution offline. The reminder application functioned quite well with a small pre-loaded cache of identity information.

2 What Is Needed for Sensing More Widely

We believe that many of the more successful pervasive applications, such as cell phones, are tightly woven into the fabric of *all* daily activities, offering continuous, mostly invisible support to literally anyone, in almost any situation, where and when the need arises. As researchers we want to explore these new opportunities, and experience living in a world where activity-sensing facilities are available in all parts of our lives, to assess the new opportunities, and their impact on human behavior. So we are trying to explore ways to deploy automatic activity sensing on a wider scale and we seek to provide *each individual researcher and their families* with technology to experience, explore, and ultimately expand how pervasive computing could impact everyday lives. We looked for solutions with the following features:

1. Low cost (<= $25), long battery life (>= 1 year), small size & weight (can be inconspicuously worn or carried in a pocket) with a consistent proximity sensing range (<= 5 meters).
2. Sensors can be deployed where they are needed: in offices, cars, homes, or even public spaces, enabling small-scale field trials.
3. Colleagues can incrementally deploy and maintain the infrastructure themselves with no requirement for centrally administered activity log or co-location database.
4. The implementation of proximity sensing should match the user's intuitive sense of proximity as closely as possible. For example, a person one floor above is not normally considered to be in close proximity.

3 Design Strategy

The most distinctive aspect of our strategy is that it aims to increase availability on a per-user basis, rather than for a whole community, or geography – to create a *personal pervasive system* technology.

Our approach employs a collection of identical lightweight portable proximity sensors, called SPECs, designed to support the kinds of tasks we described earlier. We expect our colleagues, and eventually our users to deploy SPECs themselves, dotting them around in places they frequent, attaching them to objects, or wearing them.

Fig. 1. SPECs deployed by Kyle on the garage, backpack, scooter, and himself.

Although our goal is to create a platform for investigating a range of sensing technologies, we decided that our first prototype would only sense the proximity and identity of other nearby SPECs. This "what and where" information would be captured and acted upon autonomously, or uploaded later to a server for offline analysis. To further simplify things, and inspired by Factoid [9] and Pollen [10], device discovery employs an extremely basic peer-to-peer protocol and makes no reference to a central database, or wide area wireless network.

Each SPEC broadcasts a unique 32-bit identifier (ID32) every 2 seconds (to conserve power this interval is increased automatically when the set of SPECs in proximity isn't changing). They also listen continuously for the ID32 broadcast from nearby SPECs. When a new SPEC is sighted, a sighting record is created, time stamped with the start time, and stored in a history. Each record describes an interval during which a particular ID32 was repeatedly sighted. If sightings cease for more than a specified interval (2 minutes in the current system), then the sighting record is time stamped with the end time and closed. As we shall explain shortly, the sighting history can be analyzed locally to see if the user should be alerted to any noteworthy events, or it can be uploaded, via a SPEC portal, to an internet-based service for off-line analysis, archiving, etc.

A very simple sighting history pattern recognizer is used within SPECs to detect noteworthy situations. The pattern recognizer uses a byte code form that is downloaded to the SPEC by a portal. Once downloaded the pattern recognizer runs independently within the SPEC. The pattern language is declarative and consists mainly of time interval and ID set operations. Functions are available to find particular sightings in the history. Two of these functions are called `first` and `last`. They take two parameters, an ID set and a time interval, and return the first or last sighting of any of the IDs within the time interval. Functions are composed to create reminder expressions. If the expression result is true or a non-empty ID set then the reminder is considered active. Patterns can be defined and given names using a simple XML name/value structure. For example, the pattern to detect when a SPEC with ID 3 has last been seen for more than 5 minutes in the interval from 1:00 to 2:00 is:

```
<define name="LastSeen" value="duration(last({3}, [1:00, 2:00])) > 5m"/>
```

We anticipate that more advanced applications *will* want to make occasional connections to Internet services to archive sightings, process them into a more intelligible form, invoke other actions, or download new search tasks. To do this SPEC-portals provide mobile SPECs with the means to upload sighting history. Portals also provide a time service enabling mobile SPECs to set their real-time clocks, and a means to download patterns to support real-time applications.

We have given high priority to small size and battery life and in consequence, sacrificed communications and computational power, storage and user interface capability. We aspire to achieve power budgets in the prototypes that will allow small field trials of about a week to be completed without a battery change. Transferring data to infrastructure via portals is completely optional, but does provide a means to increase the number of contexts in which sensory information can be *immediately* relayed back to an individual's database for processing.

4 "Bring It Home Again" A Real-World Reminder Application

Kyle is a sixth grader with a number of ways to go to school: bike, scooter, walk, and car ride, and a number of different things to carry with him. With all those options it's no wonder that he sometimes forgets how he got to school on any given day, or to bring something home. As a consequence, on two occasions he forgot to bring home his scooter. By the time he remembered, the scooter had been stolen! We applied SPECs to the problem.

Fig 1 shows Kyle and his SPECs. A SPEC is attached to his scooter, the wall above its parking spot in the garage, he wears one, and has one on his backpack. He placed one on his desk at school. His wearable SPEC was loaded with a pattern, shown in Fig 2, designed to notice which articles he takes to school and to remind him if he does not return the same things home.

```
<define name="Home" value="{bathroom, kitchen, garage}"/>
<define name="School" value="{desk, 'bike rack'}"/>
<define name="Things" value="{backpack, bike, helmet, scooter}"/>
<define name="School Begins" value="8:30 a.m."/>
<define name="Dismissal" value="2:45 p.m."/>
<define name="Leave To School"
 value="end(last({garage}, [,'School Begins']))"/>
<define name="Leave To Home"
 value="end(last(School, [Dismissal, ]))"/>
<define name="To School" value="['Leave To School',]"/>
<define name="To Home" value="['Leave To Home',]"/>
<reminder name="Forgotten"
 value="retain(Things, 'To School') - retain(Things, 'To Home')"/>
```

Fig. 2. Kyle's reminder pattern.

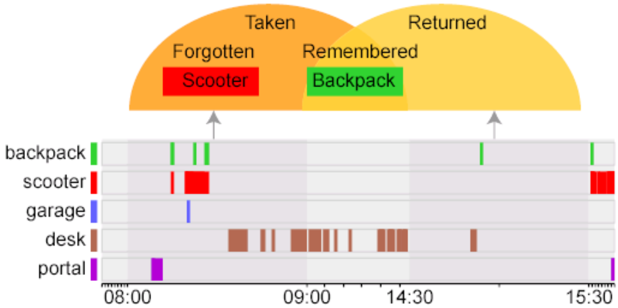


Fig. 3. Sighting records from the SPEC Kyle wore on a 'forgetful' day.

A chart of the sighting records from a forgetful day is shown in Fig 3. The various encounters can clearly be seen throughout the day: garage in the morning and afternoon, desk in between, backpack and scooter during travel times. The diagram above the chart illustrates how Kyle's SPEC keeps a watch out for any SPECs that accompany him in the period between his leaving the Garage and arriving at his Desk. In this example there are two: the backpack and scooter. The SPEC waits for school to be out *and* for Kyle to leave his desk. In this case it fails to see the SPEC Kyle attached to his scooter, the `Forgotten` reminder is triggered, and the SPEC LED starts flashing to remind Kyle he has left a tagged item behind at school. After some time, Kyle notices the reminder and returns to recover his scooter.

The tests ran for a couple of weeks. During that time he wore a SPEC the majority of the time. He constructed two different ways to wear it – as a necklace and as a bracelet. The necklace form turned out to be his favorite. It was quite a positive social experience for Kyle. Being a 6^{th} grader wearing a high tech looking gadget produces a lot of positive attention. During the test period we had a number of issues to shake out, since this was the first real use of the devices in a natural setting. We had hardware issues involving the batteries and the enclosure, in addition to software issues with acquiring the time and knowing when batteries needed to be changed.

5 SPEC Prototype

Our goal was to develop a mechanism where objects could autonomously detect each other's proximity without reference to any additional infrastructure. At this stage we were focused on low-power, easy-to-implement solutions. With our experiments in mind, we estimated that that a coin cell battery life of one week was an absolute minimum requirement for prototypes that would be used to test our ideas. However, we wanted to make sure that the basic design could be engineered for year lifetimes at some point in the future. Several different underlying technologies could be used. For our initial prototype, we only considered two *proximity detection* mechanisms: radio (RF) and infrared (IR).

The challenge for proximity detection is finding a very low-power way for devices to signal to each other. Experience with Active Badges which transmit a very short identification IR pulse every few seconds, had demonstrated that lifetimes in excess of a year could be achieved for transmission only. But base stations that receive Active Badge signals are powered from the domestic supply. In contrast SPECs, all of which are identical and potentially portable, must both send and listen, and it soon became clear that *the challenge for the power budget is continuous listening*. A simple analysis showed that from a power perspective, domestic IR, the kind used for appliance remote control, offered the best off-the-shelf solution to the continuous listening problem.

The IR detector that is currently being used, a GP1UD261XK, typically consumes 150uA. The output of the detector is low when a 40kHz carrier is detected. This signal is used to interrupt the microcontroller whenever it changes. To minimize power consumption, the microcontroller and other components sleep except when processing IR detector changes. SPECs send their ID using a 100mA IR LED using a Manchester encoded signal over a 40kHz carrier for 25ms. This is done at a rate somewhere between 2 and 30 seconds – depending on how often the set of SPECs in proximity are changing. Assuming an average rate of 15 seconds, sending consumes about 50uA on the average. We are currently using 150mAh coin cells in our prototype, which results in a lifetime approaching one-month. Stationary SPECs could use AA cells that should result in approximately a two-year lifetime.

In contrast, the best off the shelf RF receivers were in the 2mA range. Thus their power consumption is about 10x that of IR, resulting in a corresponding $1/10^{th}$ the battery lifetime.

5.1 Predictable Discovery

But low power was not the only reason for choosing IR. Considering our proposed applications, it seemed important that the SPEC's model of what was nearby should closely mimic the user's model. For example, if we wanted to recall a situation from the past, or if we wanted to set up a reminder for some future event, then the user would need to have a good model of what people, places and things the *computer* was likely to sense. With an RF-based technology the shape of the field is difficult to predict without special instrumentation, and can fluctuate unpredictability depending on what other things move through the field. So a RF-based proximity detector might detect a person, or thing in an adjacent room that was invisible to its user. This could lead to very confusing behaviors. Since IR does not pass through walls and has propagation characteristics similar to visible light, it is much easier for a user to predict what things the computer might, or might not be able to sense. However, we can recognize that there are clearly situations where being able to sense the proximity of things that you can't actually see could be very useful.

IR is has its own set of issues. Transmission follows line of sight and can be obscured by obstacles like furniture or clothing, or simply by being pointed in the wrong direction (although IR is also good at reflecting off many surfaces, such as white walls, and thus does have some diffuse characteristics also). Bright sunlight or fluorescent lights also easily confuse IR. Some IR solutions, such as IrDA, are designed for short range and are highly directional. Remote control IR detectors and emitters have components that are less directional and have greater range. Given that ease of implementation, size, and power requirements were top of our priority list, and given encouraging results using IR from other research projects, we felt remote control IR was the best compromise for our initial pragmatic prototype.

5.2 Hardware

The current SPEC prototype, shown in Fig 4, uses a PIC18F252 microcontroller and a real-time clock for time stamping observations, which it can store in 32KB of memory accessible over an I2C bus. It communicates with other nearby SPECs using a 40kHz infra-red carrier with OOK at 2666 baud and Manchester encoding giving an effective data rate of 40 32-bit words per second, and a range of 4-8 meters. The current user-interface uses a single green LED and a single button. It has auxiliary output provision for driving a pager motor, or beeper. The I2C header can be used to connect additional circuits, such as sensors. The *debug header* is used during development for loading and debugging the software kernel. Using off-the-shelf components the electronics package measures 40x15x14 mm, including two 11.6mm diameter coin cell batteries that are held in place by metal clips underneath the circuit board. Total cost is approximately $25, which includes the components, batteries, circuit board, and enclosure.

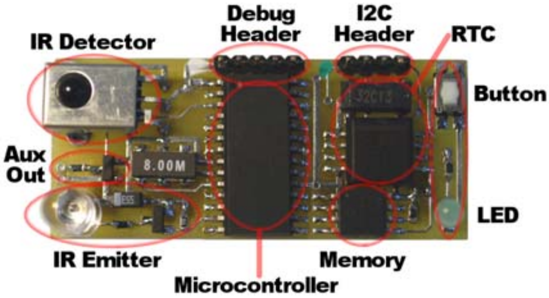


Fig 4. The current SPEC prototype board

6 Conclusions

It's a bit too early to claim that this approach to gathering context data is a success, but the results above are very encouraging. The main thing to note is *that installing the infrastructure is as simple and speedy as it sounds*, and that this approach does indeed allow us to sense more parts of our lives, including expeditions into the outside world away from work or home. Battery life allows continuous operation for useful periods. The database is optional and distributed - holding only the data for a particular set of users. We are beginning to see that SPECs can provide a valuable source of field study data for future design work and may indeed be a valuable resource for behavioral science field studies in general.

It seems that an ideal proximity sensing technology for this type of applications is currently not available off the shelf and remains an open research opportunity. The two most likely candidates suffer from having to be visible (IR) or having an unpredictable field (RF). Power consumption and battery capacities are major challenges as well.

We anticipate that our next step will be to explore using RF for proximity detection, and to add sensors to detect motion, etc.

References

1. Lamming M., Newman W. M,. Activity-based information retrieval: Technology in support of personal memory. In F.H. Vogt, editor, Personal Computers and Intelligent Systems, volume A-14 of IFIP 12th World Congress. Proceedings of Information Processing 92, pages 68--81. IFIP, Elsevier Science Publishers (North-Holland), 1992.
2. Want R., Hopper A., Falcão A., Gibbons J. "The Active Badge Location System", ACM Transactions on Information Systems, Vol. 10, No. 1. 91-102, January 1992
3. Want R., Schilit B., Adams N., Gold R., Goldberg D., Petersen K., Ellis J., Weiser M., "An Overview of the Parctab Ubiquitous Computing Experiment", IEEE Personal Communications, Vol 2. No.6, pp28-43, December 1995
4. Weiser, Mark (1991). 'The Computer for the Twenty-First Century'. Scientific American, pp. 94-10, September 1991.
5. Newman, W., Eldridge, M., Lamming, M. Pepys: Generating Autobiographies by Automatic Tracking. In Proceedings of the second European conference on computer supported cooperative work. Amsterdam: Kluwer Academic Publishers. 1991.
6. Lamming M. and Flynn M. "Forget-me-not" Intimate Computing in Support of Human Memory, Proceedings of FRIEND21, '94 International Symposium on Next Generation Human Interface, Japan, February 1994
7. Sellen, A. J., Louie, G., Harris, J. E., Wilkins, A. J. 'What Brings Intentions to Mind? An In Situ Study of Prospective Memory'. Memory, Vol. 5, No. 4, 483-507 1996.
8. Hill, J. et al. System Architecture Directions for Networked Sensors. ACM ASPLOS IX Nov. 2000 Cambridge, MA, USA.
9. Mayo, R. N. Reprint of the Factoid Page. First pub. June 1997. WRL Technical Note TB-60. http://www.hpl.hp.com/techreports/Compaq-DEC/WRL-TN-60.pdf
10. Snowdon D., et al. Pollen: Using people as a communication medium. Computer Networks, 35(4), Elsevier Science, February 2001, pages 429-442

A 2-Way Laser-Assisted Selection Scheme for Handhelds in a Physical Environment

Shwetak N. Patel and Gregory D. Abowd

College of Computing & GVU Center
Georgia Institute of Technology
801 Atlantic Drive, Atlanta, GA 30332-0280, USA
{shwetak,abowd}@cc.gatech.edu

Abstract. We present a 2-way selection method to select objects in a physical environment with a novel feedback and transfer of control mechanism. A modulated laser pointer signal sent from a handheld device triggers a photosensitive tag placed in the environment. The tag responds via a standard wireless channel directly to the handheld with information regarding an object it represents. We describe a prototype implementation for a Motorola iDEN i95cl cell phone, discuss the interaction challenges and application possibilities for this physical world selection that extends a common handheld device. We also compare this solution to related attempts in the literature.

1 Introduction

It is becoming increasingly common to carry a handheld device (a PDA or a mobile phone), with nearly always-on network connectivity and significant computational capabilities. What is not so common is the ability for these devices to facilitate interactions with the physical world. Take, for example, the development of universal remote controls for use at home. Several commercial and research projects have considered the challenge of reducing control of a large variety of home devices down to one universal platform (see, for example the recent work on the Personal Universal Controller [8], or the Gesture Pendant [13]). But in an environment with many controllable devices, it is not clear the best way to select which device should be controlled. Pointing from a distance, naturally supported by a laser pointer, seems like a good solution to this problem. We are motivated, therefore, to explore direct and natural interactions with devices in the physical world mediated through a handheld device.

A number of researchers have explored ways to tag the physical world, using printed barcodes or 2-dimensional glyphs, RF ID, or active beacons, in order to connect the physical and electronic worlds. We are particularly interested in applications of lasers because they provide a simple means of visual feedback as well as at-a-distance interaction. Furthermore, unlike any other physical world selection technique in the literature, we see the advantages of creating 2-way communication between the object in the physical world and the laser-augmented handheld. The challenge we faced was to create a practical, 2-way selection technique using a laser mounted on a conventional handheld device that provides a two-way interaction. The

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 200–207, 2003.

solution described in this paper demonstrates an augmented handheld device that communicates its identity via a modulating laser signal to active tags in the environment. These tags, logically linked to objects in the environment, can then communicate back to the handheld, establishing a two-way link. This solution allows for selection of physical objects in the environment that can then be further controlled or queried by the handheld device.

Integrating this 2-way laser pointer into a mobile phone can also enable many possibilities outside of the home. The active tags can be placed on road signs and billboards, using large tags connected to some wireless telephony service. When the billboard is spotted, the user selects it in order to receive additional information to the handset. While traveling by car, the natural visual feedback of the laser pointer might not work, so we developed a slower response vibration feedback. At an airport, these active sensor tags can be placed in the logos of the airline companies. As you walk through the airport, you can simply identify one of these tags and the gate and flight information is directly sent to the handset.

Overview

Having motivated the potential applications for this 2-way selection technique, we will next present a brief overview of existing techniques for selecting objects in the physical world. We describe the design of our laser-assisted selection technique as implemented on a commercial mobile phone using special-purpose active tags embedded in the environment. We then discuss some of the interaction challenges that influenced our design.

2 Related Work

Several researchers have explored laser pointer interaction recently, both for interaction at a distance with large displays [4, 7, 9] and also for the selection task studied in this paper [6, 12, 14]. Selection of physical objects in an environment has also been explored for various augmented reality tasks (e.g., the NaviCam system [11]). For the selection task, we identify three different approaches:

Static Tagging: A static label (e.g., a barcode or 2-dimensional glyph) is placed in the environment and read or scanned by some form of reader device [5, 10]. Barcode solutions are limited to a distance of about 1 meter [6], whereas camera-based solutions are limited only by the camera resolution and perception techniques used to decipher the glyphs. The scanning device must be connected to some service that converts the scanned information into a device identifier. This interaction is one-way and the same information is provided to every scanning device. This selection mechanism works well when it is suitable to require short-range interaction.

Camera-Tracked Laser Interaction: A popular laser pointer interaction scheme is to use a camera focused on a region of a wall or object where a laser spot may appear [1, 4, 7, 9, 14]. Simple computer vision techniques locate the red laser dot and follow it around the interaction region. Such a scheme is appealing for meetings or presentations, where you can interact with a display at a distance by simply pointing at it with an ordinary laser pointer [1, 4, 7, 9] and the XWand system demonstrates how it can be used with a collection of other sensors to support selection and interaction of devices through a special-purpose interaction device. An extension of XWand, called the World Cursor, removes the vision requirement by using the XWand to steer a remotely controlled laser pointer around a room [15]. The remotely controlled laser pointer has a model of where it is pointing in 3-space and has sufficient geometric information to know where its red laser dot is pointing. While these camera-based tracking solutions provide very flexible ways to point and select objects within the environment, they require a lot in terms of camera infrastructure and detailed geometric information and will not work well in large environments with much movement of objects.

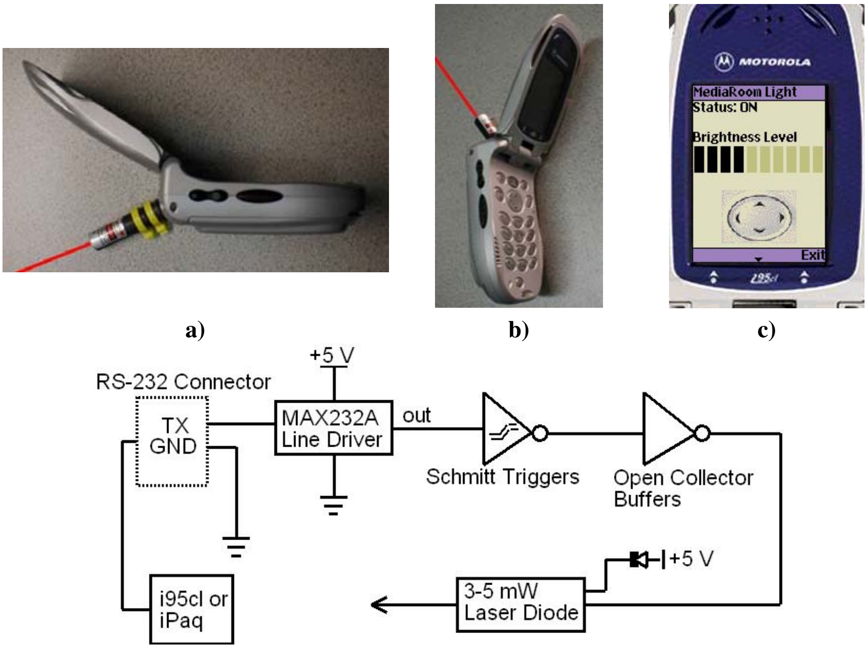
Active Tagging: Our laser pointer system, Matthias Ringwald's Spontaneous Interaction, and MIT's FindIT Flashlight use active tags that respond to an incident laser [6, 12]. A modulated laser signal encodes information that is received by the tags and decoded. The tag is active in the sense that it can respond to the initiator of the interaction with any appropriate response. The response by the FindIT Flashlight is an indicator light to notify the user that the desired object has been found. For our system, the interaction with the handheld device is 2-way because the tag can use a number of wireless mechanisms to send a data response. Ringwald's Spontaneous Interaction makes a similar attempt by sending back web content from selected tags through 802.11b Wi-Fi. While active tags are an additional expense and may require separate power, they could be placed in a variety of locations or embedded in commercial appliances. No further knowledge of the environment is required, making them a more practical solution for the selection task compared with the camera-based solution discussed above. The HP CoolTown beacons are small hardware devices distributed throughout an environment whose function is to wirelessly broadcast device references (URLs) [3]. CoolTown beacons use IRDA to broadcast device references, which make it an attractive solution for IR ready handhelds. However, IRDA lacks the precise identification and natural visual feedback possible with lasers and is limited to only a few meters in range.

3 How Laser Selection Works

Our laser system is based on a PC-to-PC laser communication link first published by GKDesign [2]. We built prototypes for a Compaq IPAQ and a Motorola iDEN i95cl cell phone. Each handheld is assumed to have wireless data access (802.11b or Bluetooth for the IPAQ and the Nextel data service for the i95cl handset), providing an IP address for direct communication. The figures and description in this paper describe the cell phone prototype because the slim form factor and always on data service motivates a variety of indoor and outdoor applications.

3.1 The Instrumented Handheld

The laser is integrated into the handheld device by using a 3-5 mW diode laser module mounted on the antenna of the i95cl handset (see Figure 1). The RS-232 line from the handset is run through a MAX232A IC line driver and an open collector buffer to allow serial-controllable modulation of the laser beam. Using the J2ME environment on the handset, it is straightforward to encode outgoing messages, and we chose to encode the IP address of the handset to facilitate routing of return messages. Since the communication with tags is asynchronous, the message from the handset is padded with start and stop codes. When the laser button is pressed (using the lower button on the handset shown in Figure 1a), the message and its padding, or the message frame, is continuously transmitted. The user only has to keep the laser shinning on a sensor tag for the length of one complete frame. At a baud rate of 9600 bps, a message frame of 128 bits would take 13 ms to be detected. We found that speeds of 9600 bps or lower required only a simple parity check for error detection; more sophisticated error detection and correction schemes (e.g., CRC or Hamming codes) would be needed at higher transmission rates or for sensitive messages. A more sophisticated message-encoding scheme could be constructed by directly modulating the laser with the Motorola chipset. This would provide larger messages and faster transmission rates. The handset's 1400 mAh lithium ion battery pack powers the laser module. The circuit draws about 35 mA when the laser is activated. Moderate use of the laser does not significantly reduce the battery life of the handset.



d)

Fig. 1. a-b) The instrumented cell phone handset, a Motorola iDEN i95cl. The antenna is angled slightly to accommodate reading the screen while pointing the laser. c) A simulator screen for the phone showing a simple control screen for an X10-controlled light switch identified through laser interaction. d) A schematic for the laser controller.

3.2 The Active Tag

The sensor end of the system consists of a bed of phototransistors, a MAX232A IC line driver, and Schmitt triggers. The actual light sensing part of the tag is a bed of NPN IR phototransistors. The signal then runs through Schmitt triggers to square up the wave before it is fed into the MAX232A line driver. Out of the line driver comes the RS-232 signal, which is fed into a microcontroller (we used a PC in our prototype). The microcontroller produces the response back to the handset.

Since the message sent by the cell phone is an IP address, the sensor knows exactly where to send the feedback information. The route of the response can vary depending on the connectivity of the sensor tag. In one prototype, we directly connected the tag to the wired LAN. In another prototype, the tag uses X10 to transmit to a basestation (an X10 server) connected to the LAN. Routing to the handset is done using the Nextel data service (or 802.11b or Bluetooth in the case of the IPAQ).

We chose phototransistors as the sensor mechanism because they are less affected by ambient light. Phototransistors are available with a variety of spectral ranges. The ones used in our sensor tags have a range of 600 nm to 1000 nm. The response range is broad enough for a typical red laser (670 nm) and still able to ignore some of the lower wavelengths.

We found there is just enough red in ambient light, especially sunlight, making a red-pass filter on the sensor ineffective at times. For even more immunity to ambient light, a black epoxy 800 nm to 1000 nm phototransistor could be used, requiring an IR laser. Since IR lasers are invisible to the naked eye, we lose the natural visual feedback that the red laser provides. Coupling a small red laser with the IR laser produces both a higher-speed transmitter and a visual feedback loop. However, IR lasers present a greater eye safety concern than red lasers.

The reason for a bed of phototransistors is to allow for a larger target. Our prototype sensor tag is enclosed in a cone structure (see Figure 2a). The inside of the structure is padded with a reflective material. When the light beam hits anywhere inside the structure the light rays are reflected in many directions. The front part of the cone is covered with a defocusing material; when the beam hits the front of the structure, the light beam is refracted in many directions inside of the cone. The bed of phototransistors resides at the back of the cone structure, increasing the likelihood of a ray hitting at least one phototransistor and reducing the need for operator precision in aiming the laser. Another advantage of using a cone-like or cylindrical structure is that very precise optics can be added so that only one phototransistor is needed. In this case a beam coming in at any point can be refracted straight to the sensor. This approach is the opposite of the defocusing used in the FindIt flashlight [6]. By defocusing at the target instead of the source, we maintain the desirable features of a laser pointer, namely the long range and natural visual feedback.

The cone-like tag is useful for long-range selection. For short to intermediate ranges, a dense bed of phototransistors is suitable and can be embedded within the objects being selected. For example, figure 2b shows a prototype of a tagged light switch.

The size of the sensor constructed depends on the placement in the environment and desired precision for the laser. More accurately, inspired by the data reported by Myers *et al.*, we calculated that the average deviation resulting from laser "wiggle" is .0025 times the distance from the target [7]. Deviation here means the amount the laser dot wiggles while a user tries to hold it steady for 3 seconds. This means a tag diameter should be at least .005 times the expected distance to accommodate easy selection. Our prototype sensor tags are 5 inches in diameter, which is large enough to interact from within a very large open space (approx. 83 feet).

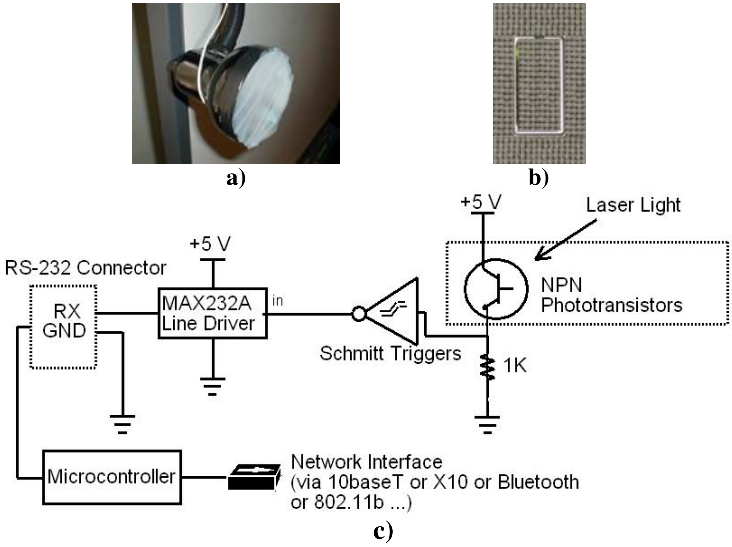


Fig. 2. a) A cone-shaped light-sensitive tag. b) A layer of phototransistors replacing a wall plate for a light switch shows a different form factor for the active tag. c) The schematic for the active tag.

4 Analyzing Handset-Tag Interaction

The interaction between handset and tag is relatively simple, with some interesting interaction challenges. The important human factors are acquisition time to locate a tag with the laser pointer, how long the user can comfortably keep the beam steady on the sensor tag to ensure a hit, and what feedback tells the user that the target is hit. Myers *et al.* showed that it takes approximately 1 second to move a laser beam to a target position [7]. This is perfectly reasonable for our scheme, since a likely alternative to pointing at a target for selection would be to look up the target device using a list on the handset, which would likely take longer than 1 second. The same study found that a laser mounted on a PDA is more stable than just holding a plain laser pointer, because the PDA provides a more effective grip because of its large size. In our experience, both the i95cl and iPAQ offer very stable control. The i95cl fits well in the palm of the hand and is the better of the two because of its intermediate size and ergonomic shape. The laser activation button resides on the left side of the phone, allowing thumb operation when held in the left hand or finger operation when held in the right hand. This allows for stable control as the laser is activated. We also angled the handset antenna to provide even more comfort and stability, allowing laser pointing and easy screen reading.

Since the dwell time for a tag to "read" the modulated laser signal is pretty low (13 ms), there are two ways a tag may be accidentally triggered. First, we know that a user cannot effectively predict where a laser pointer will hit when initially activated [7, 9], so it may accidentally hit the wrong target initially. Second, tags may be placed close to each other, as might happen in a cluster of home entertainment devices, causing inadvertent selection. One solution to this problem is to introduce a two-stage scan and select process using a two-position switch on the handset. When the button is depressed half way, the laser is turned on without any messages being sent and natural visual feedback is used to aim the laser over the appropriate target. Upon proper targeting, the handset button is depressed fully, sending out the modulated message frames. Another solution would be to deliberately slow down the active tags by requiring receipt of multiple message frames before being activated.

Another potential challenge is feedback, or knowing when a tag has made a reading. One solution, suggested by the FindIt flashlight, is to provide a LED on the tag itself that can be illuminated when the microcontroller detects a valid read [6]. Another solution, which we implemented, is to signal the handset over the air when a read is detected. The handset can respond with visual or vibration feedback. In our various prototypes, the latency for this handset feedback ranged from 700-1000 ms. The faster feedback occurred with the tag connected to the network via 802.11b wireless and the cell phone receiving feedback via its cellular network. The slower feedback was a result of using an X10 connection between the tag and the network. For many indoor applications, this latency is probably too great to be useful. However, in outdoor applications, where ambient sunlight negatively impacts the natural visible feedback of the laser, the vibration scheme is more useful.

5 Conclusions

We presented a laser-assisted 2-way selection technique for identifying and interacting with objects in the physical world. This technique uses active tags that can detect modulated signals from a handheld-mounted laser pointer and respond via some wireless route back to the handheld device. We demonstrated this prototype based on a popular mobile phone handset, the Motorola iDEN i95cl. The selection technique was originally designed for use in a home-based universal remote control application, but the use of a mobile phone with constant network connectivity opens up possibilities for this technique that would work outdoors as well as indoor.

Acknowledgements

The authors thank Motorola, and in particular Joe Dvorak, for the donation of the iDEN handsets for this research. We thank Matthias Ringwald who worked on an earlier version of laser-tracking for the remote control application. This work is sponsored in part by National Science Foundation (ITR grant 0121661) and the industrial sponsors of the Aware Home Research Initiative at Georgia Tech.

References

1. Cooperstock, J., Fels, S., Buxton, W. and Smith, K.: Reactive Environments: Throwing Away Your Keyboard and Mouse. *Communications of the ACM*, 40(7), pp. 65-73, September 1997.
2. GKDesign Engineering: RS-232 Laser Transceiver. *Electronics Australia*, pp. 56-61, October 1997.
3. HP Labs: The CoolTown Project. http://www.cooltown.com/research.
4. Kirstein, C and Müller, H.: Interaction with a Projection Screen using a Camera-tracked Laser Pointer. *Proceedings of the International Conference on Multimedia Modeling.* IEEE Computer Society Press, pp. 191-192, 1998.
5. Ljungstrand, P., Redstrm, J. and Holmquist, L.E.: WebStickers: Using Physical Tokens to Access, Manage and Share Bookmarks to the Web. *In Proceeding of Designing Augmented Reality Environments* (DARE 2000), pp. 23-31, ACM Press, 2000
6. Ma , H. and Paradiso, Joseph A.: The FindIT Flashlight: Responsive Tagging Based on Optically Triggered Microprocessor Wakeup. *Proceedings of Ubicomp 2002*, Springer-Verlag Lecture Notes in Computer Science, Volume 2498, pp. 160-167, 2002.
7. Myers, B.A., Bhatnagar, R., Nichols, J. Peck, C.H., Kong, D., Miller, R. and Long, A.C.: Interacting at a Distance: Measuring the Performance of Laser Pointers and Other Devices. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2002*). ACM Press, pp. 33-40, 2001, Minneapolis, Minnesota.
8. Nichols, J., Myers, B.A., Higgins, M., Hughes, J., Harris, T.K., Rosenfeld, R. and Pignol, M.: Generating remote control interfaces for complex appliances. *Proceedings of the 15th annual ACM symposium on User Interface Software and Technology (UIST 2002)*, ACM Press, pp. 161-170, Paris, France.
9. Olsen, D.R., Jr., Nielsen, T.: Laser pointer interaction. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2001*). ACM Press, pp. 17-22, March 2001, Seattle, Washington.
10. Rekimoto J. and Ayatsuka Y.: CyberCode: Designing Augmented Reality Environments with Visual Tags, *Proceedings of DARE 2000*, 2000.
11. Rekimoto, J and Katashi, N.: The World through the Computer: Computer Augmented Interaction with Real World Environments, *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '95),* ACM Press, pp.29-36, Pittsburgh, PA.
12. Ringwald, M.: Spontaneous Interaction with Everyday Devices Using a PDA. *Presented at the Supporting Spontaneous Interaction in Ubiquitous Computing Settings Workshop* (UBICOMP 2002). http://www.inf.ethz.ch/~mringwal/publ/ringwald-interaction.pdf. Göteborg, Sweden, September 2002.
13. Starner, T., Auxier, J., Ashbrook, D. and Gandy. M.: The Gesture Pendant A Self-illuminating, Wearable, Infrared Computer Vision System for Home Automation Control and Medical Monitoring. *The IEEE International Symposium on Wearable Computing (ISWC 2000)*, IEEE Computer Society Press, pp. 87-94, Atlanta, GA.
14. Wilson, A. and Shafer, S.: XWand: UI for intelligent spaces. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2003*). ACM Press, pp. 545-552, 2003, Ft. Lauderdale, Florida.
15. Wilson, A.: Pointing in Intelligent Environments with the WorldCursor. *Proceedings of Interact 2003, the Ninth IFIP International Conference on Human-Computer Interaction.* Zurich, Switzerland. To appear, September 2003.

Finding a Place for UbiComp in the Home

Andy Crabtree, Tom Rodden, Terry Hemmings, and Steve Benford

The School of Computer Science & IT, The University of Nottingham, Jubilee Campus
Wollaton Road, Nottingham, NG8 1BB, UK
{axc,tar,tah,sdb}@cs.nott.ac.uk

Abstract. The movement of design out of the workplace and into the home brings with it the need to develop new analytic concepts to consider how ubiquitous computing might relate to and support everyday activities in domestic settings. In this paper we present a number of concepts derived from ethnographic studies of routine activities and technology uses implicated in the production and consumption of communication in the home. These concepts sensitise design to the importance of the ecology of the domestic space and distributed arrangements of collaboration to communication. They draw attention to the places where communication is accomplished and the routines whereby communication is articulated, thereby highlighting 'prime sites' for situating ubiquitous computing in the domestic environment.

1 A New Challenge

The domestic environment is currently receiving a great deal of attention as a place for the development of ubiquitous computing. The technical and methodological challenges involved in realising computing systems for the domestic environment are significant, and require researchers to anticipate facilities that are likely to emerge in the home of the future. Anticipating the future is fraught with difficulties and researchers have tended to exploit purpose built 'living laboratories' in order to explore both the future potential and the shortcomings of current technological infrastructures, and to consider the ways in which inhabitants might experience the home of the future [3, 13, 15]. These explorations have been complemented by design led 'visions of the future' [18] that seek to convey the potential ways in which ubiquitous computing might be deployed in domestic settings. The home offers new sets of challenges that move our understanding of interaction beyond the current focus on information and knowledge work. It exposes us to the demands of new user groups, including the elderly, the disabled and the mentally impaired [14, 5], and requires us to be sensitive to the impact of broader cultural values and the need to support activities other than work [7].

A key research problem in designing for this environment is the need to understand the everyday character of the home: how people live in the home, what they do when they are at home, and the potential role of technologies within the milieu of domestic activities. This paper is concerned with the social character of the domestic

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 208–226, 2003.

environment, particular with regards to communication which is a prime area for design [11], and with developing a sensitivity to the real world, real time ways in which communications are produced and consumed in the home in order that ubiquitous computing might be woven into and resonate with domestic activities [23]. While there is a vast literature on communication in the workplace, when we turn the home we find few conceptual and analytical tools to assist design [10]. Furthermore, it has been suggested that in turning to familiar concepts derived from the workplace there is risk of migrating and operationalising a set of values that may be inappropriate for domestic design [8]. Indeed, as Gaver puts it,

> There is a danger that as technology moves from the office into our homes, it will bring along with it workplace values such as efficiency and productivity at the expense of other possibilities. (ibid.)

The problem here is that such things as ‘production’ and ‘efficiency’, which may themselves be construed of in terms of such concepts as ‘plans and procedures’ and ‘workflow’ along with a host of analytic concepts that describe the organization of practical action in the workplace in terms of Fordist and Taylorist principles of capital production, do not apply to the organization of practical action in the home. This is not to say that household members do not have a concern with the production of domestic life or with efficiency in carrying out household activities, early research in the field suggests that they do [23], but rather that such things as production and efficiency in domestic life cannot be adequately understood in the accepted terms of capital production, as domestic life is not organized in those terms by household members. The home and workplace are *different* domains and we do not need an extensive period of research to tell us that – as members of society we know that, and as designers we know it as a condition of our inquiries. The problem we have is not one of understanding that the home and the workplace are different, then, but of developing insights into how and in what ways the home is different in order that we might develop technologies that are appropriate to the setting.

Accordingly, this paper presents and articulates a number of analytic concepts that sensitise ubiquitous computing to the organization of communication in the home environment and open up the play of possibilities for design. These *sensitising concepts* have emerged from a series of ethnographic studies of domestic settings and are elaborated through the explication of an empirical instance of the collaborative production and consumption of communication in the home. We exploit the notion of ‘traffic’ as a guiding principle and consider how the primary social organizational features observed in members practical management of traffic may be exploited to develop domain knowledge and inform the development and placement of ubiquitous computing in a wide variety of domestic settings. We employ a distinct representational format to convey the overall results from our studies and to articulate a set of key properties that may be explicated by other researchers in other settings. Following this, we outline a number of challenges for ubiquitous computing to emerge from our research. First, however, we briefly consider the motivating factors behind our work, which have given rise to a distinct set of sensitising concepts for ubiquitous in the home.

2 Developing an Understanding of Technology in the Home

We offer a set of concepts sensitising design to important features of social activity and technology use in the home by reflecting on a series of ethnographic studies of the production and consumption of communication in domestic life. These studies focus particularly on what Edwards and Grinter [6] call,

> … the stable and compelling routines of the home, rather than external factors, including the abilities of the technology itself. These routines are subtle, complex, and ill-articulated, if they are articulated at all … Only by grounding our designs in such realities of the home will we have a better chance to minimize, or at least predict, the effects of our technologies.

Concurring with this position, we reason that design may be usefully informed through careful consideration of the ways in which existing technologies, whether technically sophisticated or not [25], are routinely made to be 'at home' by household members in their everyday interactions so that they come to assume an eventful and purposeful role in domestic affairs [20]. A number of ethnographic studies have already offered rich insights into the routine nature of domestic life and technology use in the home [16, 24, 26, 22]. While such insights promise to open up the design space and inform the design of computer-based technologies that support the day-to-day functioning of the household, there is a need to move beyond the particular studies of a small group of social scientists to provide conceptual and analytical tools that designers in general may employ.

Two key principles derived from previous work in the field [17] underpin the stance we have adopted in moving from the particular to the general and developing conceptual and analytical resources that ground design in the routine character of domestic life.

- Routine activities are socially organized through household members' interactions with a host of technologies distributed throughout the home, and which support the coordination of action in their use.
- The distributed character of the technologically mediated 'work' of coordination highlights the central importance of the ecology of domestic space for design.

When considering these issues, our particular interest lies in articulating the distribution of technical objects or 'media' and their placement in various shared locations around the home in relation to the flow of information involved in communication and the coordination of practical action [9]. We have exploited this point of view when studying the domestic routines implicated in the production and consumption of communication in order to identify a set of key features or properties to inform the development and placement of ubiquitous computing in the home. In the following section we present the key social organizational features to have emerged from our studies, which articulate the ways in which communication media are routinely 'made at home' by household members.

3 The Social Organization of Communication in the Home

The sensitising concepts presented here emerged from a series of long-term ethnographic studies of 22 family homes across England that began in May 2001 and are ongoing. Our concern with communication came from two principle sources.

- It was evident from our initial studies that a great deal of activity in the home is concerned with communication *coming into* and *going out of* the home, and that many of the information resources and technologies in the home are implicated in communicative action.
- Communication has been one of the major areas of development in computing generally and provides a key motivator as design moves out of the workplace and into the home [11].

To complement our broader set of studies where communication emerged as a general and pervasive issue within domestic settings, we undertook three highly detailed and focused studies of communications in three different family settings in order to develop a more coherent understanding of communicative activity and its social organization (i.e., to develop a better understanding of how incoming and outgoing communications are produced, managed and consumed). The studies actively involved the participants. Rather than have an ethnographer 'hang around' the home, we asked our participants to video communications coming into and going out of the home and to keep a log briefly describing where the communications occurred, what they were about, who was involved, and what was done in response to particular communications. This strategy had two distinct benefits.

- It meant that the ethnographers did not have to spend long periods of time waiting around for events of relevance to the research to happen. Over a one-week period during the study, incoming and outgoing communications took up around one and half to two hours of video tape. More time was spent on communication though the details were not always recorded, and understandably so, because of a variety of sensitive household matters. Nevertheless, the approach produced a rich corpus of quality data and was highly cost effective.
- Enlisting participants as data gatherers provided the opportunity to open up a highly detailed and intimate dialogue with household members. The video and logs became conversational resources which we used to explore the social organization of communication in the home in collaboration with those parties who actually do the 'work'. The approach enabled us to involve inhabitants in a design dialogue and to bring their competences to bear on design reasoning then.

Recognising that ethnography is nothing more, or less, than a data collection technique, we adopt an ethnomethodological approach to the *analysis* of ethnographic materials [4]. Ethnomethodology is concerned to explicate the social organization of activities in observable details of their local production by members, rather than in the more conventional terms of social science theory. Accordingly, we examine empirical instances of communication in theoretically unmediated detail in order that we might see what can be seen and learn what can be learnt of communication in the home as real world, real time socially organized activity.

3.1 An Instance of the Social Organization of Communication

We present a simple empirical instance collected during our studies, which should be seen and treated as an *illustrative* case that enables us to articulate key social organizational features of communication and sensitising concepts.

Dave arrives home from work and on entering the porch picks up the day's mail. He walks into the kitchen and sorts and opens the mail. One is a postcard from family members who are on holiday and one is a 'thankyou' card from a friend who stayed over at the weekend. Dave puts the cards at the front of the kitchen table for Jane to see when she gets back from work.

Fig. 1. Receiving and placing a card for the attention of others (porch and kitchen table)

Dave then goes into the living room and starts his computer, sending an email to the card sender to acknowledge receipt of the card and exchange similar sentiments.

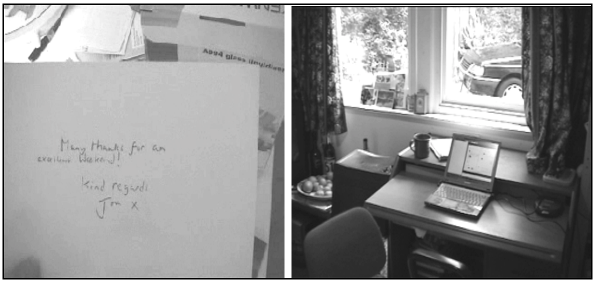


Fig. 2. Responding to the card (email from workstation in living room)

When Jane arrives home (some four hours later) she notices the cards on the table. Jane picks them up, walks into the living room where she sits down on the sofa and discusses both the weekend and family members holiday briefly with Dave. Jane then puts the card on the mantelpiece in the living room and the postcard on the windowsill.

Fig. 3. Places where cards live (the mantelpiece and windowsill)

4 Key Features and Sensitising Concepts

> **traffic** n. & v. **4** dealings or communication between people etc. **5** the messages, signals, etc., transmitted through a communications system; the flow or volume of such business. The Concise Oxford Dictionary

Treated as an illustrative case, the above instance draws our attention to several grossly observable features of communication in the home. In particular, the instance instructs us that a primary feature of the production and consumption of communication in the home is a members' concern with the practical management of *traffic* in and through the domestic space[1]. In this context it is worth noting the diversity of media used within this simple act of communication and the ways in which household members coordinate the use of a range of media as a practical part of this act. Fig. 4 represents the ecological distribution and diversity of media used in this instance.

The point of this representational *format* is not simply to represent the traffic implicated in individual instances. Rather, the purpose is to enable designers to 'get a picture' of the social organization of communication in the home as a single coherent environment. Accordingly, the role and importance of the ecological distribution of traffic and the sites at which various media are used becomes even more apparent when we consider a number of instances of communication in tandem. Fig. 5 represents communications traffic in the same household over a one-week period.

4.1 Places of Communication

The collection of instances represented in Fig. 5 moves us beyond recognition of the spatial (and temporal) distribution of communication [17] to make a phenomenon available to design reasoning that has not already been seen in the single instances, namely, the social organization of communications traffic as a whole. What emerge

[1] This view contrasts with conventional reasoning which suggests that the primary feature of communication is social cohesion [e.g. 11, 12]. Our studies do not deny the important *functions* of communication but rather, draw our attention to taken for granted organizations of communication upon which such weighty matters *rely upon or turn*.

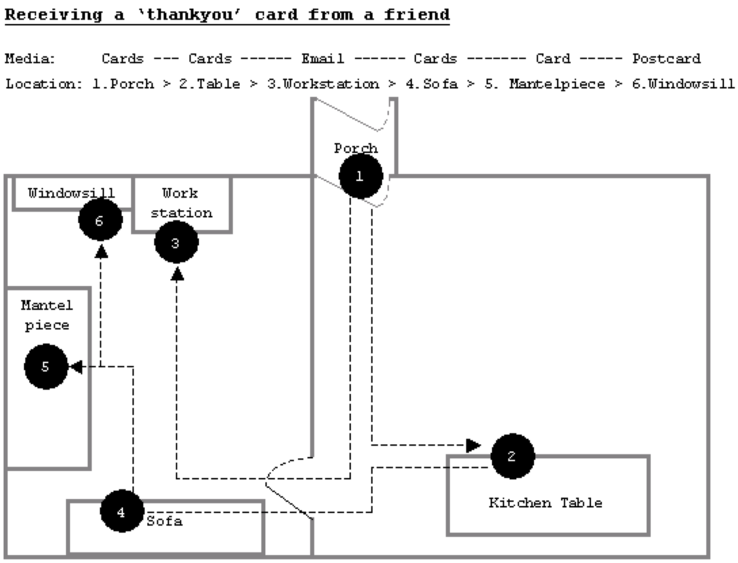


Fig. 4. Communications traffic within the home

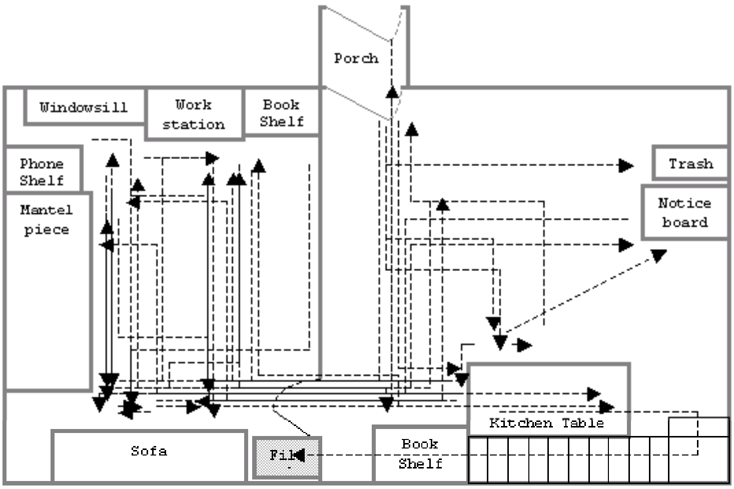


Fig. 5. Incoming and outgoing traffic over a one-week period

are patterns of technologically mediated activity and the reciprocal flow of communications traffic around the home. Thus we see traffic flow from familiar *place* to familiar *place* in the home. In the rest of this section we consider the social organization of traffic in terms of the familiar places implicated in communications coming into and going out of the home. In particular, we consider three different, grossly observable properties of these places and there relationship to the production, management and consumption of communication. These include:

- **Ecological Habitats:** places where communication media live and where residents go in order to locate particular resources.
- **Activity Centres:** places where media are actively produced and consumed and where information is transformed.
- **Coordinate Displays:** places where media are displayed and made available to residents to coordinate their activities.

In the rest of this section we explicate each of these in turn, moving from textual description of the single instance to the representational collection in order to make the social organization of communication in the domestic environment available to design reasoning and inspection in other residential settings.

Ecological Habitats. One of the initial observations to emerge from our studies is that the various media implicated in communication live in particular places. Household members do not have to search for the mail, or the computer, or the telephone and address book, etc., because they *situate* communication media in particular places from where they may be readily retrieved or accessed when they are needed. As a rule, or matter of routine use, communication media live in particular places where they may be readily located. This, of course, is not to say that communication media do not stray, that members do not lose things. Indeed, such occurrences demonstrate the rule as it were and may be accounted for by invoking the ordinary notion of *mis*-placing things. More formally, we might call these places 'ecological habitats'. The term draws analytic attention to the physical surroundings within which communication media reside. Ecological habitats are readily available to observation. They are in plain view and require no special methods to see. In the case of our illustrative instance, the windowsill and mantelpiece are employed as ecological habitats (places where certain kinds of card live). The workstation (from where the responding email is sent), the kitchen table (where the card is placed for the attention of others), and the sofa (where the cards are used), each elaborate different organizational properties, which will be addressed below. First, however, if we consider the representational collection the following ecological habitats become visible across the home as a whole.

Each of these habitats was illuminated or made visible by a single instance (sometimes recurrently, where mail or the phone was involved for example) and through the use of particular media, which the single instance elaborates in detail.

Activity Centres. The places where communication media live (ecological habitats) are not necessarily the same places where communication media are used and our observations highlight this. The instances shows, for example, how on being noticed (through their placement at the kitchen table) the cards migrate to the sofa where they become conversational objects for Jane and Dave. We call the places where communication media are used 'activity centres' insofar as the collection of instances shows us that there are certain places in the home where communication media are recurrently employed. The collection of instances reveals the following activity centres in the study setting.

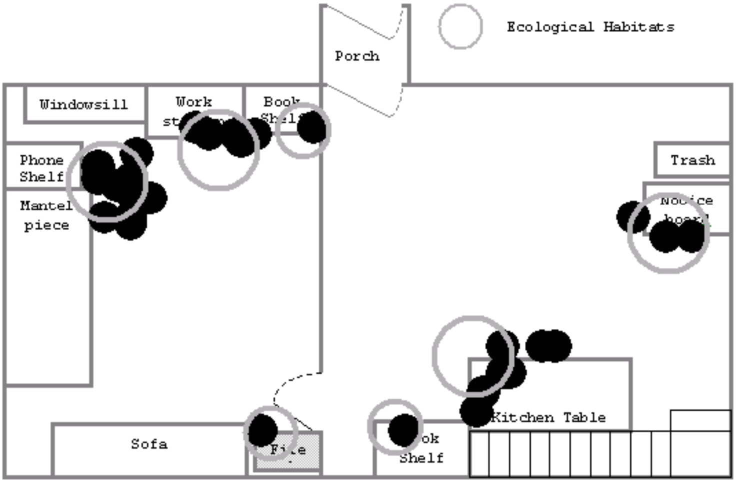


Fig. 6. Ecological Habitats (Phone Shelf, Workstation, Noticeboard, Kitchen Table, File, etc.)

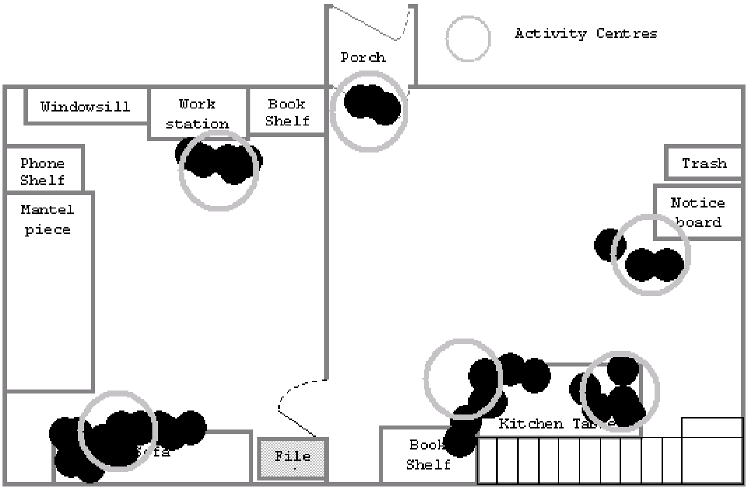


Fig. 7. Activity Centres (Sofa, Workstation, Porch Noticeboard, Kitchen Table)

Although distinct, some ecological habitats and activity centres overlap (see Fig. 6 for comparison). In this home, the workstation, noticeboard, and kitchen table are at different times employed by members to perform different roles. For example, the kitchen table is at one time a habitat for the handling of mail while also being a centre for conducting phone calls. The noticeboard is at one time a habitat for information of short-term relevance (appointment cards, concert tickets, school term dates, etc.) and at another a centre where the information situated there becomes a resource in social interaction. Similarly, the workstation is at one time a habitat where documents are kept and displayed as reminders of ongoing jobs of work and at another a centre where emails are received and sent. This overlap is important for reasons that will be articulated in due course.

Coordinate Displays. Household inhabitants routinely construct displays from out of the flow of communication media. On receiving the cards, for example, Dave places them on the kitchen table for the attention of Jane, who subsequently places them on the mantelpiece and windowsill. The mantelpiece and windowsill are ecological habitats that are employed to display certain objects in this home and while interesting insofar as such sites provide a 'home' for future display technologies (such as electronic picture frames), what is of more interest to us is the construction of what we call 'coordinate displays'.

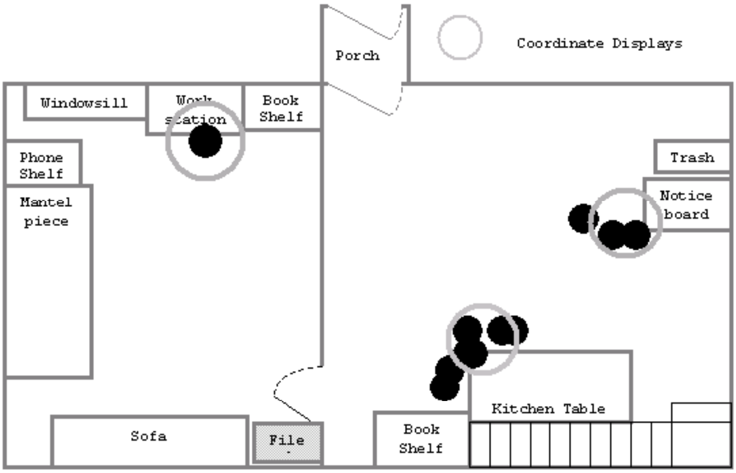


Fig. 8. Coordinate Displays (Workstation, Kitchen Table and Noticeboard)

The kitchen table provides us with a prime example of a coordinate display. The kitchen table is a tacitly agreed upon site between Dave and Jane for the placing of mail. Both know that incoming mail will be placed at this site. No words or discussion is needed as they can see at-a-glance that mail has arrived that requires attention by the very act of its visible placement and display. The important point about such sites of display is the rationale and function of their construction. The display of mail not only triggers conversation between household members as with the cards, but more importantly, it triggers practical action such as the timely paying of household bills, renewing vehicle tax or household insurance, for example, *not that the person who opens the mail is necessarily the who takes action*, however. In other words, the construction of displays at certain sites through the placing of mail (and other media implicated in communication) provides for the coordination of practical action. As is the case with ecological habitats and activity centres, these places are architecturally and aesthetically contingent. In other words, their specific location depends on the particular physical layout of the home and the arrangement of 'mobilia' or furniture therein. Just as some places may at one time serve as activity centres and at others as ecological habitats, then so too they may also serve as sites for the construction of coordinate displays. For example, the noticeboard is at one time a place where information of short-term relevance is kept to-hand, at another a place where that information is employed as a resource in communication (coordinating family visits through consulting school term dates, for example), and at another time a place where the information placed there displays and so provides for the timely coordination of social activities (such as taking the children to a party, attending a dentist's appointment, or paying an invoice at the end of the month). Just what places overlap and serve multiple functions will depend on the particular residential environment under study.

4.2 The Flow of Communication

The various places outlined above only make sense as part of an overall flow of communication. A key property organizing the flow of communication identified in our studies is that of discrete and recurrent 'sequences of action'. Sequences of action consist of the routine courses of action and technology use that link the various ecological habitats, activity centres and coordinate displays together in any single instance. Dave and Jane's handling of the cards shows, for example, the recurrent ways in which such media are routinely handled in their home: through placing cards on the table and in different locations where they are used and subsequently live (see Fig. 4). Other instances in this particular home show how the handling of mail more generally is organized through the recurrent construction of coordinate displays at the kitchen table and noticeboard and how mail of long-term relevance migrates to particular habitats (such as the domestic filing system). Individual sequences of action *elaborate the social organization of particular forms of communication in particular settings*. For example, on getting up in a morning, someone might collect, open and read mail at-the-kitchen-table over breakfast, placing items to be dealt with later on the bureau-in-the-living-room and bills to paid on-the-trolley-in-the-hall. Sequences of action emphasize how information is spatially and temporally distributed throughout the home. They draw our attention to the particular ecological habitats, activity centres and coordinate displays implicated in particular forms of communication in particular settings and convey to designers the everyday routines of the home. The routines embodied in sequences of action are known to the inhabitants of the house and are used as a resource for managing their activities and for handling communication. Thus items to be 'worked' upon are placed within an appropriate and routine sequence of action. Thus, and for example, the packed lunch is left on the kitchen table where correspondence is opened and read in the morning, or beside the porch where household members may place various media to be taken to work. In the following section we consider the design implications to emerge from our consideration of the social organization of communication in the home.

5 Putting Sensitising Concepts to Work

Close examination of a corpus of empirical instances has enabled us to identify a set of socially organized features or properties that constitute a *system of communication*. Recognition of this socially organized system, and the ways in which inhabitants manage traffic through discrete and routine sequence of action that link various locations in the home together, provides us with a set of sensitising concepts that may be employed by designers to find a place for ubiquitous computing in a wide variety of residential settings. Obviously everyone's house is different - a broad set of architectural and aesthetic contingencies are involved in the layout of the home. Nevertheless, members of the architectural community have already highlighted stable ways in which people set up and configure the spaces they occupy. For example, work on patterns presents common arrangements [1], while work on the evolution of buildings highlights the underlying dynamics of change in the 'space plan' of the home [2, 19]. To complement these insights, the concepts we offer seek to convey the ways in which different ecological features of the home are exploited by members to manage and coordinate domestic activities. The point is not that every home will have a kitchen table and that bills are kept there in order that they can be found and acted on appropriately. Indeed, many homes may not have a kitchen table or may not even have a separate kitchen at all, especially in non-Western cultures. Nevertheless, we would suggest that *each home will have its own* ecological habitats, activity centres and coordinate displays that are constructed, arranged and linked together by household members in the course of carrying out the routine sequences of action whereby the produce, manage and consume communications.

Culture – whether understood in terms of nationality or age, gender, and identity, etc., is manifest in the routine character of communicative action in the home and the role of particular locations or places implicated in such action. In other words, 'culture' is not something separate from communication, something that stands behind it and shapes it as it were, but visibly implicated and manifest in its ecological organization. Accordingly, we suggest that there is a need for designers to be aware of the ecological character of communication and to chart the various places 'at work' in communication in order that ubiquitous computing might resonate with and so fit into domestic life in a wide variety of different settings. Such a sensitivity might be developed by conducting short periods of ethnography (whether through direct immersion of a fieldworker or the administration of household members) in order to gathering a corpus of single instances. These instances may be examined for their organizational properties:

- Firstly, for the routine sequences of action implicated in technology usage, no matter how technically sophisticated that technology may be.
- Secondly, for the ecological habitats, activity centres, and coordinate displays that are elaborated by routine sequences of action.
- Thirdly, findings may be represented to the design team through the use of the representational format to explicate the system as a whole.

Elaborating these organizational properties serves to convey a rich portrait of how communications are produced and consumed within a particular setting and orients the development team to a particular set of design concerns. In the following section we reflect on how the emergent concepts outlined above may be used to drive the design and deployment of ubiquitous computing for the home.

5.1 Placing Technologies in the Home

Considerations of the nature of the domestic space and the relationship and placement of technology therein is already of major concern to ubiquitous computing. Researchers have suggested that design will be required to develop a wide range of media spaces to support domestic communication [11]. Others have explored the integration of sensing technologies and digital services within the domestic space [13]. We seek to provide conceptual and analytic resources for the research community that will help guide the placement of ubiquitous computing to meet the routine day-to-day needs of inhabitants and so situate new and emerging technologies in appropriate places in the home. The need to integrate media spaces and digital services with the architectural and aesthetic fabric of buildings is emphasised by the notion of 'roomware' [21]. Roomware consists of such components as the *DynaWall* (an interactive electronic wall), *CommChairs* (mobile and networked chairs with integrated interactive devices), and the *InteracTable* (an interactive table). The relationship of new and emerging technology to the arrangement of domestic space has also been explored through the use of Pattern Languages and seen the emergence of the notion of *comZONES* [12]. As with roomware, this use of patterns is predicated on the integration of the digital into *new, purpose-built environments*. Consequently, it is not at all clear how existing approaches support the 'fitting' of technology into pre-existing environments in the piecemeal fashion that has been predicted for the adoption of ubiquitous computing in the home [6]. The sensitising concepts we have provided assist designers and help them to address this problem by orienting designers to the different ways in which particular locations are routinely employed and so 'situate design in the home' by elaborating the various places in particular environments that provide candidate locations for future technologies.

Prime Sites for Technology. One of the most obvious uses of the concepts we have provided is to highlight 'prime sites' for ubiquitous computing in domestic settings. Our approach makes visible the ways in which a host of technologies are ordinarily employed. This in turn supports the identification of the ecological habitats, activity centres and coordinate displays associated with a particular setting and so provides a resource with which to frame design. For example, it has already been noted that some ecological habitats, activity centres and coordinate displays overlap, as can be seen below.

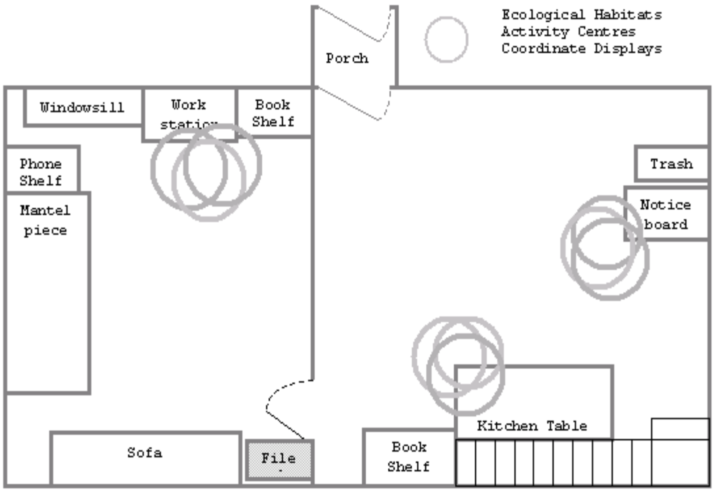


Fig. 9. What Place Might Ubiquitous Computing Find in the Home?

The places of overlap highlighted in Fig. 9 identify locations that inhabitants habitually exploit in the handling of communication. They draw attention to locations that users routinely return to in order to manage communication within the home and, consequently, identify places that offer good candidate locations for placing ubiquitous computing in particular settings. Their explication allows designers to reflect upon the nature of these overlaps within particular environments, contrasting the ways in which digital functionality is currently concentrated at the desk in the living room, for example, with the openness and flexibility of the noticeboard and the kitchen table to open up the play of possibilities for design. If we were to consider extending digital functionality across this household through the implementation of a *DynaWall* and *InteracTable*, for example, then the points of overlap elaborated above suggest that this would be best achieved by placing those technologies in the kitchen to create a network of digital services and surfaces manifest in locations that Dave and Jane habitually exploit to accomplish communication.

The Convergence of Media. Our concepts also highlight the diverse collection of media that are used by inhabitants in carrying out the daily routines involved in managing communication. In designing systems to fit into these settings we often need to consider the different forms of media that new technology will have to find a place alongside. Our studies suggest that rather than displacing existing media in the home, new technologies are used alongside a variety of different media that are employed across a range of different sites. The diversity of media involved in the household study reported here and the places where they are manipulated is reflected in Fig. 10.

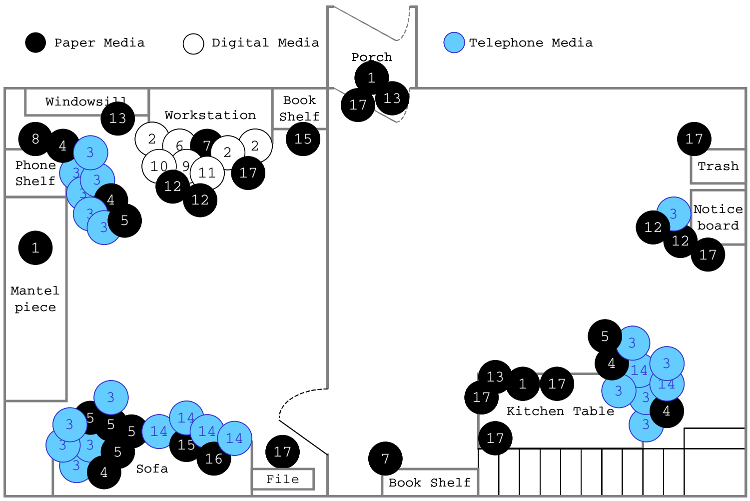


Fig. 10. Media Usage across Ecological Habitats, Activity Centres, and Coordinate Displays

Key to Fig 10.

1. Cards
2. Email
3. Phone
4. Address Book
5. Paper note
6. Internet
7. Paper recipe
8. Answer machine
9. Electronic documents
10. Hyperlinks
11. Digital images
12. Paper documents
13. Postcard
14. Text message
15. Book
16. Magazine
17 Mail (bills etc.)

Essentially, this representation draws attention to the various sort of media that coalesce at particular places and allows designers to pose questions as to whether or not they seek to supplement, augment or replace existing media. Furthermore, given the collection of empirical instances, this supports the making of design decisions with some definite insight into the ways in which new technology might impact upon current organizations of communication in the home.

More generally, this particular representation makes it perspicuous that paper-based media are well integrated into the home environment. Paper-based media can ‘find a home’ in any ecological habitat and coordinate display (they can be put in drawers, left on surfaces and pinned to walls). The means of creating and modifying them can be easily used in any activity centre (you can write and draw in activity centres throughout the home). Digital media, by way of contrast, are less comfortably integrated. Some, such as email, Internet and hyperlinks don’t easily spread beyond the workstation, which is still required to produce, manage and consume them. To break this dependence, inhabitants often transform digital media to paper by printing them out. They also leave paper pointers to digital media, writing notes to remind others to read an email from a friend, for example. The mobility of devices may impact upon this. Telephone media are more widely spread throughout the home, for example. However, such mobile media still tend to cluster around but a few locations where they coalesce with other media (such as address books and paper notes), which suggests the need to *link* ubiquitous computing with other media at these places.

5.2 Building on Communication Places

In addition to raising a set of pertinent questions regarding the places where new and future technologies might be situated to meet the day-to-day needs of particular households, our research has provided a set of concepts that can be matched to existing and emergent research agendas. The three main features of places of communication provide a conceptual guide for more targeted investigation that combines more focused studies with different forms of technological development. Essentially, in just the same way that the concepts associated with the workplace (task, role, privacy and workflow, etc.) allowed researchers to develop research agendas within HCI and CSCW, then our concepts may be used to motivate and illuminate research questions in ubiquitous computing. In this section we wish to provide a brief illustrative example of research issues to emerge in our own work from each of the different features of place.

- **Ecological Habitats** are places where communication media reside. They are places where users return to find the resources needed to deal with communication activities. As we have seen in our studies, digital media currently tend to be closely connected with digital devices. In contrast, paper finds its way to a greater variety of places and uses in the home. What might it mean to make the digital more prominent throughout the home? How might the presence of various media - particularly non-digital media - be represented in ecological habitats to allow them to be digitally available? How might we mange issues of security and privacy when ecological habitats are made digitally available? These and a host of other issues, including the digital evolution of ecological habitats, represent interesting areas of future study.
- **Activity Centres** are places where media are manipulated, consumed and transformed. These places provide a key set of research issues regarding the augmentation of existing media used at them and beg the question as to what new forms of device may be developed for activity centres? Might we use electronic displays to augment electronic noticeboards or calendars, for example? How may a system represent the ‘work’ that goes on in activity centres to household members in order to support the management of activities within the home? How may a system make activity centres available at a distance, particularly from outside the home? How may a system exploit knowledge of the ‘work’ carried at activity centres and support access and privacy? Are sensing technologies a solution and do activity centres provide a guide to place video cameras and to guide video recognition to identify media uses and interactions that occur there?
- **Coordinate Displays** are places where communications media are made available to others in the domestic setting in order to support the coordination of activities. Primary research issues surrounding coordinate displays focus on recognising the events to be coordinated, and the various media implicated in coordination, to consider how these are best propagated throughout the household. It might also be important to consider how can we augment coordinate displays to make the information displayed available outside of the domestic setting? If so, the representation of

information and associated issues regarding the management of distributed collaborative access and control are important matters here and present significant challenges to the design of new technologies that merge the digital with the physical fabric of the home.

5.3 Exploiting Sequences of Action

As a final reflection we wish to briefly consider the routine sequences of action that elaborate and link ecological habitats, coordinate displays and activity centres together. These sequences are the means by which communication is handled and they articulate the sensitizing concepts we have presented. Designing technology to support sequences of action raises a number of questions, many of which arise from the limited penetration of digital media. Essentially digital media need to be more flexible in terms of how individual items are moved to and from ecological habitats, manipulated in different ways at activity centres, and placed to be seen by others at coordinate displays. Sequences of action raise distinct design questions regarding how devices may be used to support the distributed flow of objects and information around the home. For example, how might we assign email to various locations as with paper mail? Or again, how might personal devices be used to coordinate actions within sequences of action, enabling individuals to see and pick up email when it has been left in a public place for them? Sequences of action are not only topics for design, as it were, but also raise questions as to how they may be used as a resource within applications to support the overall management of communication in the home. This is a different order of question that shifts the focus from one concerned with the use of technology to one concerned with supporting the activities involved in sequences of action. Accordingly we might ask how we might exploit representations of sequences of action to make information available to others when it is most relevant? For example, might a system exploit the sequence of action associated with the principle bill payer to best *place* a reminder to pay a bill as he or she leaves for work? Or again, how might we exploit representations of sequences of action to monitor coordinate actions across the household, providing notification that a bill payment has been made by associating the payment with a prior sequence of action? These and the other questions articulated above open up a host of complex research issues. They may be explored in a range of different settings and further elaborated by other researchers through continued ethnographic study and the application of our sensitising concepts to the empirical materials gathered.

6 Conclusion

This paper has argued for the need for new conceptual and analytic tools to inform the development of ubiquitous computing as design moves out of the workplace and into the home. We have developed a set of sensitizing concepts from careful consideration of the empirical material drawn from a range of ethnographic studies of routine activities and technology uses in domestic settings. These concepts makes visible the socially organized production and consumption of communication in the domestic environment and sensitise design to the importance of the ecology of the domestic space and distributed arrangements of collaboration to communication. In particular, they draw design's attention to the key properties of ecological habitats, activity centres, and coordinate displays. Through routine sequences of action these are the places where communication 'gets done' and where a host of different communication media are manipulated and used. They highlight 'prime sites' for the placement of ubiquitous computing and elaborate a set of design questions informing the development of existing and emerging research agendas. The concepts we have presented are currently being employed to understand the unique requirements of a variety of very different homes, ranging from family environments to residential care settings, and to determine the functionality to be provided by new technologies as they are placed in a number of homes.

Acknowledgement

This research was funded by the Equator IRC (EPSRC GR/N15986/01), www.equator.ac.uk.

References

1. Alexander, C. (1979) *A Timeless Way of Building*, New York: Oxford University Press.
2. Brand, S. (1994) *How Buildings Learn*, New York: Viking
3. Brumitt, B. et al. (2000) "EasyLiving", *Proc. HUC '00*, pp. 12-29, Bristol: Springer.
4. Crabtree, A. (2003) Designing Collaborative Systems: A Practical Guide to Ethnography, London: Springer-Verlag.
5. Crabtree, A. et al. (2002) "Sore legs and naked bottoms", *Proc. of Dependability of Computer-based Systems*, London: The Royal Statistical Society.
6. Edwards, K. and Grinter, R. (2001) "At home with ubiquitous computing: seven challenges" *Proc. UbiComp '01*, pp. 256-272, Atlanta, Georgia: Springer.
7. Gaver, W., Dunne, A. and Pacenti, E. (1999) "Design: cultural probes", *Interactions*, vol. 6 (1), pp. 21-29.
8. Gaver, W. (2001) "Designing for ludic aspects of everyday life", *ERCIM News*, No.47. www.ercim.org/publication/Ercim_News/enw47/gaver.html
9. Harper, R., Evergeti, V., Hamill, L. and Strain, J. (2000) *Paper-mail in the Home of the 21st Century*, Digital World Research Centre. www.surrey.ac.uk/dwrc/papers/okios.pdf
10. Hindus, D. (1999) "The importance of homes in technology research", *Proc. CoBuild '99*, pp. 199-207, Pittsburgh: Springer.
11. Hindus, D. et al. (2001) "Casablanca: designing social communication devices for the home", *Proc. CHI '01*, pp. 325-332, ACM Press.
12. Junestrand, S., Keijer, U, and Tollmar, K. (2000) "Private and public digital domestic spaces", *International Journal of Human Computer Interaction,* vol. 54 (5), pp. 753-778.

13. Kidd, C.D. et al. (1999) "The aware home", *Proc. CoBuild '99*, pp. 191-198, Pittsburgh: Springer.
14. Mynatt, E., Essa, I. and Rogers, W. (2000) "Increasing the opportunities for aging in place", *Proc. CUU '00*, pp. 65-71, Arlington, Virginia: ACM Press.
15. Mozer, M. (1998) "The neural network house", *Proc. AAAI Symposium on Intelligent Environments*, pp. 110-114, Palo Alto, California: AAAI.
16. O'Brien, J. and Rodden, T. (1997) "Interactive systems in domestic environments", *Proc. DIS '97*, pp. 247-259, Amsterdam: ACM Press.
17. O'Brien, J. et al. (1999) "At home with the technology", *ACM Transactions on Computer-Human Interaction*, vol. 6 (3), pp. 282-308.
18. Philips Design (2000) *Visions of the Future*, www.design.philips.com/vof
19. Rodden, T. and Benford, S. (2003) "The evolution of buildings and implications for the design of ubiquitous domestic environments", *Proc. CHI '03*, pp. 9-16, Florida: ACM Press.
20. Sacks, H. (1992) "A single instance of a phone-call opening", *Lectures on Conversation* (ed. Jefferson, G.), Lecture 3, Spring 1972, pp. 542-553, Oxford: Blackwell.
21. Streitz, N.A., Geißler, J. and Holmer, T. (1998) "Roomware for cooperative buildings", *Proc. CoBuild '98*, pp. 4-21, Darmstadt, Germany: Springer.
22. Tolmie, P., Pycock, J., Diggins, T., Maclean, A. and Karsenty, A. (2002) "Unremarkable computing", *Proc. CHI '02*, pp. 399-406, Minneapolis: ACM Press.
23. Venkatesh, A. (1985) "A conceptualization of household-technology interaction", *Advances in Consumer Research*, vol. 12, pp. 189-194.
24. Venkatesh, A. (1996) "Computers and other interactive technologies for the home", *Communications of the ACM*, vol. 39 (12), pp. 47-54.
25. Venkatesh, A. and Nicosia, F. (1997) "New technologies for the home", *Advances in Consumer Research*, vol. 24, pp. 522-528.
26. Venkatesh, A., Stolzoff, N., Shih, E. and Mazumdar, S. (2001) "The home of the future", *Advances in Consumer Research*, vol. 28, pp. 88-96.

New Perspectives on Ubiquitous Computing from Ethnographic Study of Elders with Cognitive Decline

Margaret Morris, Jay Lundell, Eric Dishman, and Brad Needham

Proactive Health, Intel Research, JF3-377, 2111 NE 25th Ave., Hillsboro, OR, 97124
margaret.morris@intel.com

Abstract. A rapidly growing elder population is placing unprecedented demands on health care systems around the world. Cognitive decline is one of the most taxing health problems in terms of both its relation to elders' overall functioning and the cost of care. The needs of elders with cognitive decline - for invisible, intuitive support and assessment – invite a reconsideration of the assumptions behind and specifications for ubiquitous computing solutions. This paper describes findings and implications of ethnographic research conducted with cognitively impaired individuals and their informal care networks in 45 households in 5 U.S. regions. Key themes regarding needs and barriers to successful aging are addressed through a set of design principles which apply across the stages of cognitive decline. To convey stage-specific findings and associated challenges for ubiquitous computing, case studies of four representative households and example concept solutions are presented. The design principles and technology challenges outlined in this paper may generalize to other contexts for ubiquitous computing.

1 Introduction

"A friend of mine called the other day ... he had a good day because he bought a car. I said I had a good day, too, because Betty made coffee on her own."

- Bill, husband of an Alzheimer's patient

"Gerry always used a computer, but he has a difficult time using it now."

- Alice, wife of a dementia patient

When robbed of the ability to use tools as basic as a coffee maker due to a disease such as Alzheimer's, people are forced to rethink their everyday priorities and assumptions about how they will interact with the world. Bill's struggle is unfortunately typical of those faced by the millions who care for elders with declining capabilities and consequent lifestyle changes. And like Gerry, who cannot remember what he has learned late in life, many can no longer interact with relatively recently acquired and novel home devices, such as computers and remote controls.

Cognitive decline may well invite reconsideration not only among sufferers but also the ubiquitous computing community. In particular, concepts such as ubiquity, adaptivity, contextual awareness, location-based services, and usability may take on new meaning. What might be learned about general ubiquitous computing principles

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 227–242, 2003.

from extreme cases such as advanced Alzheimer's households? And how might ubiquitous and proactive systems improve prevention, early detection, and caregiving for people dealing with cognitive decline?

The two examples above are far from unusual, as the incidence of cognitive impairment is rising along with the dramatically increasing lifespan. Cognitive decline is part of old age to varying degrees – in fact, it is estimated that 50% of those over age 85 meet criteria for Alzheimer's Disease. Over the next five decades, the incidence of Alzheimer's alone is expected to rise from 4 to 14 million in the U.S [1]. From mild decline to severe dementia, cognitive impairment is one of the biggest threats to independence and quality of life [2]. Cognitive decline is one of many health care problems faced by the growing elderly population that invite novel paradigms of care [3, 4, 5]. Ubiquitous computing for home based health care may help reduce the unprecedented strains these health problems are placing on medical resources, and hopefully also allow elders to live with greater independence and enjoyment.

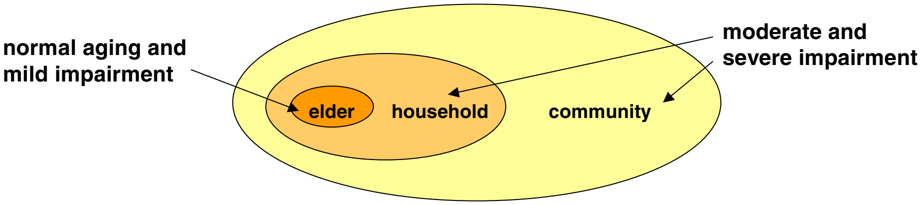


Fig. 1. Audiences for technologies. As cognitive impairment becomes more severe, the audience for technologies extends from the elder to caregivers in the household and community.

To understand the needs brought about by cognitive decline and to begin developing a computing paradigm that will support sufferers and their caregivers, Intel's Proactive Health Research team conducted in-depth ethnographic fieldwork with households coping with cognitive decline and healthcare professionals specializing in dementia. A common set of needs emerged across the stages of cognitive impairment. These needs and their implications for technology solutions are outlined in four design principles. Concept solutions geared towards specific stages of impairment are presented as illustrations of these principles. The solutions proposed range from those geared towards improving the quality of life for elders with mild impairment to those aimed at alleviating the burden experienced by caregivers of elders with moderate to severe dementia. All concept solutions were designed with the goal of prolonging independence, giving elders the option of "aging in place."

2 Related Work

Following is a discussion of several projects of direct relevance to the current study. For the sake of brevity, we have not included a comprehensive review of technology aids for the elderly.

A number of "Smart Home" platforms have been developed to implement and assess ubiquitous technologies. Several prominent examples include MIT's house_n project [6], the Georgia Tech Aware Home [7], and the Center for Future Health at the University of Rochester [8], which features an intelligent medical advisor and early detection capabilities, such as gait assessment. The Gloucester Smart House [9], is specifically targeted to support people with dementia with relatively simple solutions, such as a bath and basin monitor to prevent overflows, and a locator for finding lost items such as keys and glasses. Rather than observing people in a lab home, Honeywell has deployed its Independent Lifestyle Assistant (ILSA) system in actual homes to test their activity detection and messaging system [10].

Other researchers have developed specific devices to provide broad based health assistance to elders. Pollack [11] has developed and tested a cognitive orthotic – the “Nursebot” – that prompts elders through activities of daily living and guides them through their environment. At the International Center on Technology for Successful Aging (ITCA), the cell phone is a focal tool for location-tracking and contextual prompting of elders and their caregivers [12]. In a more specific application area, Tran and Mynatt [13] have developed a memory aid for cooking in which cameras record events and display snapshots to show elders where they are in the cooking process.

Oatfield Estates, a residential care facility in Oregon, monitors and tracks residents via infrared badges and sensors throughout the living facilities [14]. This facility is unique in that it can easily incorporate and assess new technology in a real living environment. Intel is working with Oatfield to identify new opportunities as well as barriers to utilization of technologies. In addition, two of Intel’s collaborators, Misha Pavel at Oregon Health Sciences University and Henry Kautz at the University of Washington, are using sensor data from this site to test statistical inferencing about behavior.

3 Extending Previous Research

Like the projects above, Intel is exploring location and activity tracking technologies to support elders’ independence and support safety monitoring. Our objectives extend, however, to enhancing quality of life through technologies that will facilitate social connectedness and continuation of meaningful activities. Significant differences also lie in our needs gathering and prototyping approaches. We take an ethnographic approach, immersing ourselves in elders’ routines and environments, with the belief that “aging in place” solutions will require a deep understanding of elders’ everyday behavior and relationship to the home. Additionally, the solutions we develop are “experience prototypes” designed to swiftly probe the user experience in the home environment.

3.1 Procedures

The ethnographic needs inquiry, situated in the research plan shown above, was designed to illuminate topics that might be hard for elders to self report – their values for successful aging, everyday struggles and rituals, and resources for coping with the challenges of cognitive decline. The tiered data gathering approach began with focus groups that surfaced a breadth of themes, and was followed by contextualized household interviews and intensive observations to explore particular themes in greater depth. Data collection was conducted by social scientists on Intel's Proactive Health team in collaboration with university researchers.

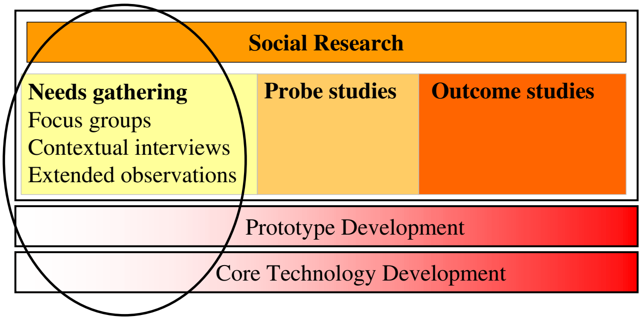


Fig. 2. Ethnographic needs' gathering is the first phase of the Social Research track, which develops concurrently with Technology and Prototype Development. Future in-home Probe studies will drive concept iteration [15] and Outcome evaluations will assess the livability and effectiveness of prototypes.

Focus groups – structured conversations with specific target groups – were conducted with elders and spouses, family caregivers and medical caregivers. Topics of inquiry included values for successful aging, challenges of everyday life, strategies for remembering and organizing, variability in cognitive functioning, life changes since the onset of cognitive decline, coping resources, interests and valued activities, and attitudes about "aging in place." Family and medical caregivers were also asked how they detect decline and day-to-day variability, and how they adjust care accordingly. The focus groups not only surfaced important themes, but were also a mechanism to select participants for home interviews. Focus groups were audiotaped for further analysis.

Contextualized interviews were conducted in the home with all available members of a household; they lasted between two and three hours. Interviews began with a discussion of lifestyle, history of illness, resources and concerns. Interviews included tours of the home and a review of daily routines. We observed as participants demonstrated their use of high and low technology tools. Collaboratively, we mapped out participants' social networks and timelines of precipitous events leading to health and lifestyle changes. Interviews were documented with videotape and digital photos.

Several extended observations ("shadows") were also conducted. One researcher spent several days living with an Alzheimer's patient and another day following a home health care nurse who treats dementia patients. These observations were documented with extensive notes and digital photographs.

3.2 Participants

Participants were selected from five regions: New York, Florida, Oregon, Washington, and California. Interviews and focus groups were conducted with cognitively impaired elders, caregiver spouses, family members, and professional caregivers. We recruited participants through collaboration with university researchers. Participants' involvement in other clinical trials afforded some diagnostic information.

In total, we conducted 45 household interviews[1], and seven focus groups. Of the household interviewees, ten were healthy aging elders, seven suffered from mild cognitive impairment, twenty-five were in various stages of dementia (ranging from mild to severe), and three were family caregivers of deceased dementia patients. Elders and their spouses and/or other family members participated in the household interviews. Of the focus groups, one was with healthy elders, two were with mild cognitive impairment elders and spouses, two were with dementia patients and their spouses, and two were with professional caregivers. Approximately ten individuals participated in each focus group, including some couples and some individuals. Our participants (not including children-caregivers) ranged in age from 56 to 97. The majority of our interview participants lived with a spouse or romantic caregiver, but some lived alone. Households ranged from urban, to suburban, to rural, with a mix of socioeconomic status; nontraditional households (siblings, friends, unmarried couples) and elders living in senior living environments were also represented.

4 Principles to Guide Ubiquitous Computing for Cognitive Decline

From our qualitative analysis, key themes emerged regarding unmet needs that could be addressed through ubiquitous computing. These needs – for early detection, assistance that adapts to functional variability, and social connectedness, all through comfortable interactions with familiar devices – are addressed in the four principles below.

4.1 Supporting Early Detection with an Awareness of Denial as an Obstacle: Embedded Assessment with a Strong Value Proposition for Everyday Living

Research finding: Across all stages of impairment, we observed elders and caregivers struggling with a tension between a desire for early detection on the one hand and strong denial on the other. This tension was rarely explicitly expressed, but consistently emerged in participants' narratives. All wished they had received forecasting (diagnosis and prognosis) that could have guided healthcare choices and helped them avoid crises and extend periods of relative independence. In retrospect, though, almost all acknowledged having overlooked initial symptoms, sometimes for years. This denial was driven by uncertainties and fears about dementia, as well as optimism regarding the possibilities of illness. This is a delicate issue since optimistic denial regarding the prospects of future illness can be helpful. Indeed, households without this optimism seemed to surrender independence prematurely with a kind of learned helplessness [16]. Nonetheless, in most cases denial leads to a reactive, crisis-driven style of caregiving that negates opportunities for early detection and treatment.

[1] This includes several caregiver interviews conducted in Stockholm, Sweden.

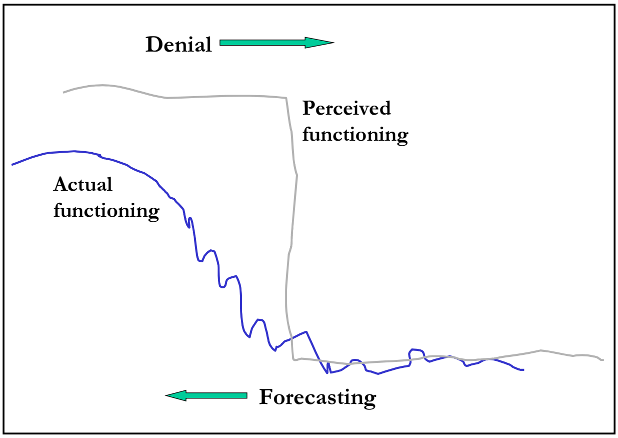


Fig. 3. Denial often delays awareness of cognitive decline and leads to reactive rather than proactive health care choices. Elders tend to overestimate their cognitive functioning up until a catastrophic event and afterwards underestimate functioning. Technologies to assist with early detection, such as embedded assessment tools, need to also address factors driving denial and offer clear value propositions for everyday functioning. The model above extends a diagram adapted by Hirsch et al. [17] to describe elders' self-perception of general health functioning.

Implication: Technologies to assist with early detection and monitoring of performance over time will need to address the factors driving denial, such as fears and uncertainties, as well as optimism. Such technologies will generally be most effective if put in place before or shortly after the onset of impairment. In order to overcome resistance, these technologies will have to have practical value in everyday life. An example is "embedded assessment" – personalized analogues of standardized cognitive tests that are embedded in elders' everyday routines and environments. Such tools could enhance recall of items such as tasks and medications or, as explored in a prototype below, recognition of names and faces.

4.2 Adapting to Variability in Functioning to Offer the Optimal Level of Assistance

Research finding: There is tremendous individual variability in the course of cognitive decline. Furthermore, for many individuals, there is considerable variability in the degree of impairment over the course of a day, a week, and the longer phases of illness. For example, one Alzheimer's patient we shadowed [18] was able to navigate complex hiking trails with ease and speed, but became completely disoriented in his neighborhood supermarket. We encountered other reports of individuals in the advanced stages of dementia having sporadic hours of great lucidity.

Implication: Adaptable technologies are needed to accommodate the lifestyles and struggles of different individuals. Even more challenging, technologies should adapt to the fluctuations in any particular elder's functioning. In the short term, caregivers may have to place a support system into different modes that are appropriate given the shifting needs of the person suffering from dementia. Longer term, reliable inferences about an elders' current functional state might allow ubicomp systems to automatically adjust the level of support. Such inferences would be based on data from sensors and cognitive tests embedded into everyday activities such as puzzles, games or manipulation of home interfaces.

Feedback about functional variability could also help individuals gain mindfulness about their patterns of mental lucidity. The benefits of mindfulness are well demonstrated for a wide range of health problems [19] and they should extend to cognitive impairment. Ideally ubiquitous computing systems would encourage individuals to leverage their periods of greatest lucidity to accomplish tasks such as planning and organizing that may not be possible during moments of lower functioning. For the severely impaired, variability information could be valuable to caregivers – alerting them to when the patient is typically at greatest risk and perhaps even identifying contextual triggers related to lucidity and confusion.

By tracking and adjusting to variability in functioning the system can offer elders the optimal amount of assistance. In general, it is best to enable users to do as much as possible in the home – exercising control over the environment is stimulating and fosters confidence and optimism [16]. Excessive or premature assistance can invite passivity and functional decline. An example from our fieldwork is a woman who preemptively moved into an assisted living environment, began receiving help with basic skills such as bathing and now lacks the drive to do these tasks on her own. For the cognitively impaired elder the benefits of controlling one's own environment need to be carefully balanced against the risks of injury.

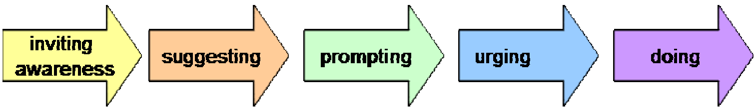


Fig. 4. Continuum of assistance. Support should escalate from simply guiding attention, to subtle suggestions, to explicit instructions, to ultimately acting on behalf of the elder.

4.3 Catalyzing Social Relationships

Research finding: Social connectedness has been shown to protect against dementia and most certainly ameliorates the pain of cognitive decline for both afflicted elders and their caregivers [20]. In our research, socially isolated elders appeared less well on a host of dimensions relating to cognitive, emotional and physical health. In addition to its intrinsic value, socializing is a strong motivation for participation in other healthy behaviors, especially exercise. Regarding the nature of social connectedness, most elders want to feel that they are having an impact on others rather seeing themselves as passive recipients of help.

Implication: Technology systems should aspire to catalyze rather than replace human interactions. One avenue is to help people share information with others in their social network in a way that invites timely communication. As explored in a prototype below, ambient displays illustrating physical and social activity levels and other health related information may motivate friends and relatives to call upon each other for exercise, companionship and other forms of socializing and support. Ubiquitous systems that offer the possibility of new forms of connectedness which become as important to health as medical diagnostic and biosensor devices.

4.4 Leveraging Familiar Interfaces

Research finding: We observed that elders tended to avoid designated "computer rooms" and instead used "command centers" (a kitchen table, a favorite chair in the living room, or for the severely impaired, a bed). Unless technologies were in easy reach of these command centers, they generally were not used. Most cognitively impaired individuals struggled to use computers, even if they had significant experience with them in the past. Future elders may have more sustained usage of PCs than today's elders, given their longer exposure to them. But even for this cohort, the complexities of computer usage may eventually pose problems. Familiar and highly tangible interfaces, particularly those that draw on procedural rather than declarative knowledge, will probably have the longest usability. Research reviewed by Lezak [21] has shown that, despite many deficits, Alzheimer's patients can often still learn simple motor skills and sometimes hold onto the ability to perform enjoyable activities they learned in the past. Indeed, several elders in our study with severe limitations in language, attention, and memory were still able to play piano or interact with some tangible interface, such as a photo album.

Fig. 5. Potential computing interfaces.

Implication: The effectiveness of the proactive health offerings suggested in this paper – early detection, adaptive support, and social facilitation – will depend on familiar, unintimidating interfaces. Computing needs to draw on the devices elders currently use in their everyday routines. Our research suggests a number of everyday surfaces and tools for interactive computing, such as bureau tops, refrigerator doors, mirrors, watches, hearing aids, TVs, and remote controls. Research with tangible user interfaces – especially with RFID tagged objects like photos or everyday objects – may allow elders with dementia to interact with computing systems much longer and more effectively than they could with screen-based systems.

5 Case Studies and Corresponding Technology Prototypes

Following are four case studies that illustrate the struggles associated with different stages of cognitive decline[2]. For each case, we describe a component of an experience prototype system that could address the problems faced by these elders.

[2] The names have been changed to preserve confidentiality.

5.1 Healthy Aging

Aging is associated with some cognitive decline, most notably slowed information processing and deficits in working memory [22]. When these cognitive changes are added to the commonly experienced declines in sensory and motor functions, independence and quality of life can be compromised. These challenges are exacerbated by the social isolation that typically accompanies aging.

Stimulation – social, mental and physical – has demonstrated value in preventing cognitive decline [20, 23, 24]. Technologies that enable elders to continue the activities that make them feel most engaged and most connected with their social networks could have powerful health benefits.

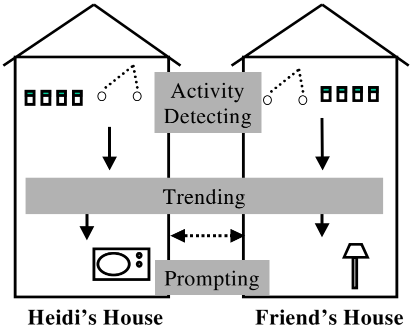
A Case Study: Heidi. At 75, Heidi is physically fit and socially active. Like many of today's elders, she takes her health seriously and assertively explores treatment options. She receives a variety of health research newsletters that keep her up to date on issues such as the role of vitamins and hormones in preventing dementia. In recent years she has been troubled by memory concerns, occasionally forgetting names, appointment dates or directions.

Heidi used to be an avid hiker and would frequently enjoy daylong explorations by herself. On a couple of occasions she lost her way and ended up in a dangerous situation. She now feels safer sticking to well known routes and always going with a particular woman she met through a hiking group. This companionship allows her to get more exercise than she otherwise would and she enjoys the conversation. She also feels that she is helping her new friend stay in shape. However, Heidi is sometimes hesitant to call her new hiking partner because she doesn't like to burden people. She wishes she knew when it was a good time to call.

Experience Prototype to Support Healthy Aging: Facilitating Social Connectedness. A prototype to address Heidi's needs monitors physical activity and facilitates communication between exercise partners. The monitor tracks both partners' movement inside and outside the home, trends it over time, and signals opportune moments for joint activity to both partners. The trending and prompting information would appear anywhere that is appropriate – the TV screen, a clock radio or even a wind chime. This same "opportunity hunting" technology principle could be used to facilitate other forms of social connectedness, such as helping people know when it might be a good time to ask someone else out for coffee, a meal, or to engage in a phone conversation.

Technology Ingredients. This prototype consists of activity tracking sensors that detect and log activity in the household, an inference engine for detecting activities and deciding on appropriate responses, and an intelligent network of household objects and appliances for interacting with elders. In our current prototype, 3D tracking is implemented via infrared cameras that detect an infrared beacon worn by the elder. Multiple cameras triangulate on the infrared beacon to determine location within the household. Movement throughout the house is logged into an activity database. When the elder goes out for a walk, a pedometer logs the steps and upon return home, syncs up with the activity tracking system (this part of the system has not yet been implemented). The inference engine monitors the activity of the elders and signals them at opportune moments to contact each other. In the current prototype, a wind chime is used to signal the elder's need and availability for exercise, but any interface could be used.

Need: Heidi relies on companionship to motivate her exercise.



Experience prototype: Activity tracking and ambient displays that connect exercise companions.

Fig. 6. Need and experience prototype for healthy aging.

We connect the sensors with motes – small, relatively inexpensive wireless processors that can be positioned nearly anywhere and automatically configure themselves into an ad hoc network [25]. They can be placed in a home without having to run wires or carefully place the devices in accessible locations. Motes can easily be connected to different types of sensors and programmed to respond in a variety of ways to input. One drawback of motes is that they run on batteries, which must be replaced regularly. Current research is focused on making them smaller and less power intensive.

5.2 Mild Cognitive Impairment

Individuals with Mild Cognitive Impairment have significant memory decline but do not meet clinical criteria for dementia. A percentage of these individuals go on to develop dementia; others improve or remain stable [26].

A Case Study: Ben. Ben, a 76 year old man who lives with his wife, Lee, was diagnosed with Mild Cognitive Impairment one year ago. Of his current problems, he is most distressed by his inability to recognize business clients. Ben painfully described the day when he could not recognize a long-time friend and client who walked into his office. Ben has few outside interests – his life has always revolved around his work and his close relationships with his clients, many of whom are also his friends. He lives in fear of the day when he will have to give up the business that has provided him with his core identity.

Experience Prototype for Mild Cognitive Impairment – Easing Name and Face Recognition. A prototype to address Ben's needs is an embedded assessment and rehearsal tool that would allow mildly impaired individuals to practice name and face recognition[3]. The primary goal of this tool is to ease social anxiety, and as a consequence social connectedness, by aiding person recognition. A second goal is for timely and ecologically valid assessment data that could facilitate early detection of further decline. This assessment would be presented through actionable feedback about variability in performance (e.g., that Ben's recognition tends to be better in the morning, and therefore mornings might be a good time to make social visits) or suggestions of strategies for recognizing people that compensate for memory decline. In its current form, this concept would not be appropriate for elders with dementias that prevent learning of new material, such as Alzheimer's, or for those with severe short-term memory impairment.

This concept invites the elder with mild impairment to collaborate with a spouse, co-worker, or friend in developing a digital timeline and rich media database of important people, places and events. The database can include photos, video clips, and voice recordings. To practice, the elder selects tangible photos which link up with the database. Questions about the person in the photo are presented on a TV screen or other interface. If unable to recognize the person in the photo, the elder would be presented with helpful cues (such as a voice clip, or a reminder of how the individual fits into the user's social network).

Need: Ben struggles to remember the names of his friends and clients.

Experience prototype: Embedded assessment and name-face rehearsal tool.

Fig. 7. Need and experience prototype for mild cognitive impairment.

Technology Ingredients. This prototype includes RFID tags in photos that represent social contacts for name and face recognition. The objects are read using an RFID reader, and the RFID data is used to access a media database. The information from the database is displayed on a contextually appropriate output device, i.e., the television located in the same room as the elder. The application progressively reveals hints and related information from the media database according to the user's needs during a particular session. Data from each session, such as the number of hints required and the response time, are stored and trended over time.

---

[3] This is currently a partially working demo, not a prototype.

5.3 Moderate Cognitive Impairment

Moderate dementia is characterized by severe memory loss and significant functional impairment. The functional impairment often makes it impossible for these elders to live alone. Due to profound trouble with language, fine motor control, and performance of sequential behaviors, they struggle with routine tasks such as cooking, and may leave out critical steps, such as turning off the burner.

A Case Study: Betty. Betty is a 59-year-old woman with early onset and rapidly advancing Alzheimer's. According to her husband Bill, Betty was previously very bright and had a remarkably broad vocabulary. After a series of stressful events in Betty's life, her family began to notice changes: She was forgetful, less articulate and had difficulty with simple computations. When shopping one day, her daughter was alarmed to see Betty unable to count change. Bill took Betty to a psychologist, and after a series of tests and re-tests, she was diagnosed with Alzheimer's disease. That was three years ago.

Today, Betty has profound difficulty with memory, language, and perception. She often struggles to remember the name of her two-month-old granddaughter, and is unable to read a clock face or piano sheet music. One of Betty's most notable challenges is in carrying out routine sequential tasks. In our interview she demonstrated her struggle to prepare a cup of coffee. With palpable anxiety, she tentatively retrieved her cup from the cupboard, opened the coffee can, put in the coffee filter, and spooned in the coffee. She turned on the coffee machine, and nothing happened. She had forgotten to pour the water into the machine.

Experience Prototype for Moderate Cognitive Impairment – Adaptive Coaching through Sequential Routines. Our prototype addresses Betty's need for help completing sequential routines. It focuses on tasks related to hydration – a need that is commonly neglected among the cognitively impaired. The system detects the location of the elder and delivers contextually appropriate prompts that escalate from abstract suggestions to explicit directions to get a drink. Through RFID and motion detecting, the system follows the user into the kitchen and provides prompts if he or she loses track of tea preparation steps. Adaptive prompts, given only when a step is missed, appear on the kitchen TV or another familiar interface.

Technology Ingredients. This prototype consists of multiple types of sensors, an activity inference engine, and prompting devices. The infrared system described in the healthy aging case is used to track general location and orientation within the house. RFID tags in the shoes of the elder are detected when the elder passes through doorways or steps on rugs that contain RFID readers. Simple switch sensors can determine when cupboard doors have been opened, and sensors in cups and containers in the kitchen determine what the elder is doing. All of the sensor data flows into an inference engine that determines what activity is being performed, and where the elder is in the process. It must also determine if the elder has been interrupted, and identify appropriate times to intervene. Prompting devices are everyday objects such as TVs, radios, lights, etc. An important aspect of this system is that it allows the elder to be in control whenever possible. It does this by tracking the elder's general abilities and

Need: Betty forgets the next step as she tries to prepare coffee.

Experience prototype: An instrumented kitchen that tracks activity and prompts as necessary.

Fig. 8. Need and experience prototype for moderate cognitive impairment.

their current level of functioning. For example, if the elder usually requires prompting to make tea, but is having a particularly good day, the prompts might be delayed.

5.4 Severe Cognitive Impairment

Severe cognitive impairment is characterized by decline in everyday functioning that renders the elder heavily dependent on caregivers. In addition to cognitive decline that drastically limits communication abilities, coordination and perceptual difficulties necessitate help with activities of daily living (such as eating, bathing and dressing).

A Case Study: Stan. Stan is a 67-year-old man with late stage dementia. He lives with his wife Nancy (age 63) of 40 years. Nancy first started to notice that something was wrong with Stan about ten years ago. He seemed forgetful and had difficulty finding the right words to express himself. To Nancy, this was very unusual, particularly since Stan had always been highly verbal. She ignored these signs initially, partly because she was dealing with her own father's Alzheimer's disease at the time.

Stan eventually saw a doctor, but the tests were inconclusive. It was clear he had memory loss, but not whether it was permanent or a result of depression. Eventually, Stan was diagnosed with Lewy Body dementia, a disease with different cognitive and motor characteristics than Alzheimer's. Stan has now lost most of his ability to talk; his speech is mostly limited to playful babbling. Every morning, Nancy wakes him, showers and dresses him, feeds him breakfast, brushes his teeth, and gets him ready for the day. Stan likes to sit in front of the TV and watch videos. Nancy still works at home as a free-lance writer. As Stan's condition has worsened, he is at greater risk of falling and Nancy hesitates to leave his side.

An Experience Prototype for Severe Cognitive Impairment – Ensuring Safety. Our prototype addresses the need for the elder's safety and the caregiver's freedom through a pervasive sensor and activity tracking system, coupled with contextualized, adaptive reminders. This system detects when Stan is up and at risk of falling, where to alert Nancy, and with what degree of urgency.

Need: Nancy is afraid to leave Stan alone for fear he will wander and fall.

Experience prototype: Chair sensors and location trackers can detect when Stan is in danger and notify Nancy.

Fig. 9. Need and experience prototype for severe cognitive decline.

Technology Ingredients. The prototype system for severe cognitive impairment consists of a sensor network, an inference engine, and a prompting system similar to the one above. Pressure sensors in the household chairs determine when someone is sitting in them, and RFID tags in the elders' shoes combined with an RFID reader in the rug in front of the chair identify the specific person. The inference engine detects activities, and determines when a prompt is appropriate. In this case the caregiver, not the elder, is prompted when there is a potentially dangerous situation. The system needs to track both the elder and the caregiver, and to determine when it is necessary to intervene. For example, it might not intervene if the elder stands up from his chair when the caregiver is in the same room, but would send a notice to an output device if the caregiver is in another room.

6 Conclusion

Through ethnographic research, we have learned about the lifestyles, aspirations, and unmet needs of a rapidly growing population. Their primary unmet needs – for early detection that is empowering and actionable, adaptable and enabling support, and facilitation of social connectedness, all via comfortable, overlearned interaction with familiar devices – can drive design of innovative solutions for cognitive impairment, other health issues, and conceivably a much broader array of problems. The findings from this study also broaden our expectations of ubiquitous computing. For example, ubiquity in these households will require a wide palette of networked touchpoints, not only of typical computing devices (e.g., PCs, laptops, PDAs, and tablets) but also of everyday devices (e.g., alarm clocks, radios, and televisions). Adaptive systems, will have to adjust not only to differences among individuals, but to the variability in each person's functioning, over short and longer periods of time. To be contextually aware, systems must identify not only the elder's location within the home, but also the proximity and positioning with regard to potential touchpoints, the amount of physical, social and mental activity in and outside the home, and even the quality of that activity. Finally, usability should be conceived of as livability; it is not enough that elders can use a system – they must want to do so routinely. If necessary, the elder's network of caregivers will also need to interact with the system through private, secure channels. An ongoing challenge is to design experience platforms that meet these requirements and function in the context of real lives and real homes.

References

1. Jorm, A.F., Jolley, D: The Incidence of Dementia: a Meta-Analysis. *Neurology*. Vol. 51 (1988) pp. 728-733
2. Agüero-Torres, H., Thomas, V. S., Winblad, B., Fratiglioni, L.: The Impact of Somatic and Cognitive Disorders on the Functional Status of the Elderly. Manuscript.
3. Coughlin, J.F: Technology and the Future of Aging. *Proceedings. Technologies for Successful Aging. Supplement to Journal of Rehabilitation Research and Development*. Vol. 38 (2001) pp. 40-42.
4. Dishman, E., Matthews, J. T., Dunbar-Jacob, J.: Everyday Health: Technology for Adaptive Aging. *National Research Council Workshop on Adaptive Aging*. January 23-24, 2003
5. Greenspan, A: Aging Population. Testimony before the special committee on aging, U.S. Senate, February 27, 2003.
6. Intille, S.S.: Designing a Home of the Future. *IEEE Pervasive Computing*, April-June 2002, pp. 80-86.
7. Kidd, C. D., Orr, R. J., Abowd, G. D., Atkeson, C. G., Essa, I. A., MacIntyre, B., Mynatt, E., Starner, T. E., Newsletter, W.: The Aware Home: A Living Laboratory for Ubiquitous Computing Research. *Proceedings of the Second International Workshop on Cooperative Buildings* - CoBuild'99. Position paper, October 1999
8. Marsh, J.: House Calls. *Rochester Review*, Spring-Summer 2002 Vol. 64, No.3, pp. 22-26
9. The Gloucester Smart House. http://www.bath.ac.uk/bime/projects/smart/index.htm
10. Haigh, K.Z., Geib, C.W., Miller, C.A., Phelps, J., Wagner, T: Agents for Recognizing and Responding to the Behavior of an Elder *Proceedings of the AAAI Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care*, Edmonton, Canada, July 2002. pp. 31-38
11. Pollack, M. E., Engberg, S., Matthews, J. T., Thrun, S., Brown, L., Colbry, D., Orosz, C., Peintner, B., Ramakrishnan, S., Dunbar-Jacob, J., McCarthy, C., Montemerlo, M., Pineau, J., Roy, N.: Pearl: A Mobile Robotic Assistant for the Elderly. *Proceedings of the AAAI Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care*, Edmonton, Canada, July 2002
12. Mann, W., Helal, S., Smart Phones for Elders: Boosting the Intelligence of Smart Homes. *Proceedings of the AAAI Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care*, Edmonton, Canada, July 2002

13. Tran, Q., Mynatt, E.: Cook's Collage: Two Exploratory Designs. Position paper for "*New Technologies for Families" Workshop*, CHI 2002, April 22, Minneapolis, MN.
14. Donahue, B.: Byte, Byte Against the Dying of the Light. *Atlantic Monthly*, May 2001.
15. Hutchinson, H., Bederson, B.B., Druin, A., Plaisant, C., Mackay, W., Evans, H., Hansen, H., Conversy, S., Beaudouin-Lafon, M., Roussel, N., Lacomme, L., Eiderbäck, B., Lindquist, S., Sundblad, Y., Westerlund, B: Technology Probes: Inspiring Design for and with Families. To appear in CHI 2003.
16. Seligman, M. E. P.: Helplessness. W. H. Freeman and Company, San Francisco. 1975
17. Hirsch, T., Forlizzi, J., Hyder, El, Goetz, J., Stroback, J., Kurtz, C.: The ELDer Project: Social, Emotional, and Environmental Factors in the Design of Eldercare Technologies. *Proceedings of the conference on universal usability*, 2000. Arlington, Virginia
18. Salvador, T., personal communication.
19. Langer , E.: Mindfulness. Addison-Wesley, Mass. 1989
20. Fratiglioni L, Wang HX, Ericsson K, et al. The Influence of Social Network on the Occurrence of Dementia; a Community-based Longitudinal Study. *Lancet*; Vol. 355 (2000) pp. 1315-19.
21. Lezak, M.: Neuropsychological Assessment, 4^{th} ed, in press.
22. Park, D., Schwarz, N.: Cognitive Aging: A Primer. *Psychology Press*, 1999
23. Yaffe, K., Barnes, D., Nevitt, M., Lui, L., Covinski, K.: (2001). A Prospective Study of Physical Activity and Cognitive Decline in Elderly Women. *Archives of Internal Medicine* Vol. 161 (2001) pp. 1703-1708.
24. Wang, H.X., Karp, A., Winblad, B., Fratiglioni, L.: Late-Life Engagement in Social and Leisure Activities is Associated With a Decreased Risk of Dementia. *American Journal of Epidemiology*. 1081-1087 (2000).
25. Kahn, J. M., Katz, R. H., Pister, K. S. J.: Next Century Challenges: Mobile Networking for "Smart Dust." *Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking*, pp. 271-278, August 15-19, 1999, Seattle, Washington.
26. Petersen, R.C., Smith, G.E., Waring, S.C., Ivnik, R.J., Tangelos, E.G., Kokmen, E.: Mild Cognitive Impairment: Clinical Characterization and Outcome. *Arch Neurology* Vol. 56 (1999) pp. 303-308

Practical Considerations of Context for Context Based Systems: An Example from an Ethnographic Case Study of a Man Diagnosed with Early Onset Alzheimer's Disease

Tony Salvador and Ken Anderson

Intel Corporation, 2111 NE 25th Ave (JF3-377), Hillsboro, OR, USA, 97124
{tony.salvador,ken.anderson}@intel.com

Abstract. The meaning of context with respect to computational systems has been the focus of considerable discussion specifically as related to context aware and proactive computing. However, there are no reports of people's direct, experiential understanding of the "lived experience" of context. As a result, there is a significant gap between theoretical approaches for understanding context and the actual practice of context, which is critical for the specification of systems. This paper reports the results of an ethnographic case study that illuminates the practical nature of context and highlights specific challenges for ubiquitous computing systems in general. We conclude that context is simultaneously more subtle, fluid and idiosyncratic than previously reported under theoretical approaches to understanding context. We further suggest implications for the design of computing systems based on these findings.

1 Introduction

In computer science, several researchers have discussed theoretical approaches for understanding context [1–6]. In recent work Dey et al, [4] assert a definition of context that accounts for various abstract states of context. For example, they define context as "…typically the location, identity and state of people, groups and computational and physical objects". That is, while they might account for *variables* they do not) account for the actual *values*. Greenberg further argues that context is a dynamic construct "…a dynamic construct as viewed over a period of time, episodes of use, social internal goals and social influences". Knowing the appropriate variables, e.g., "location", Is but one part of what's required for context aware systems. The appropriate "values" must also be known. And they must be known sufficiently for the system. The question posed to us, then is this: what are the values and under what conditions can they be known to the system?

Like context, much of human culture is often "in the background" infusing everything people do in their daily lives. Ethnography, literally "the writing of culture", was developed as a way to represent culture, informed heavily by the often unvoiced underpinnings of societies. Over the last decade, ethnographic techniques, such as participant observation, shadowing, contextual inquiry, etc., have been adapted and applied to a wider variety of settings, including more recently business settings and

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 243–255, 2003.

questions arising from other academic disciplines, e.g., aging [6]. In this paper, we present our use of these techniques in a case study to understand the lived experience of context [7].

We suggest that *values* far out weigh the importance of the *variables* in the design of context aware systems, although it is the variables that have been previously discussed in the literature. Specifically, based on the case study described herein, we suggest that context is often qualitatively subtler than has been previously suggested, more fluid and more idiosyncratic – often simultaneously. Further we suggest that it is not merely a matter of more research or more time but that the many – often simultaneously occurring - *values* that define context at any particular moment are fundamentally unknowable in anything but the most constrained situations. We hasten to add that we have not abandoned ship. Rather, we should make some course adjustments, taking into account what we do know and differently supporting situations in which the proactive understanding of context may be an untenable ideal. We offer some possibilities of this at the end of this paper.

1.1 Background to the Case Study

As we've stated, context is often heavily "backgrounded", that is, it's tough to put context into high relief such that we can know it when we see it. In these situations, a useful technique is to identify contrasting environments from which one can more readily discover what's fairly well hidden.

In this case, as part of a much wider project related to context aware computing, we were fortunate to meet "Bob" a 57-year-old male diagnosed with early Alzheimer's Disease, with whom I (the first author) spent two full days "shadowing" his every move (see Methods below). As part of a previous project, we had interviewed Bob and his wife, Carol, in considerable depth and thus had quite a lot of relevant background knowledge about Bob and Carol before we began our two-day shadow. In addition, we have been able to set Bob and Carol's observations within the context of numerous other observations through out the general course of the wider project.

Bob was diagnosed with early onset Alzheimer's at 55 years of age. There is a strong family history of the disease from his father's side and both he and his wife (Carol) were sensitive to indications of the disease. As a result of some early, albeit isolated, incidents, both Carol and Bob harbored suspicions. A few years later, he underwent genetic testing, which revealed the presence of the late onset APOE-4 gene from his father, which was expected, but also from his mother, which was a surprise. Having genes from both parents appears to change its expression from late onset to possibly early onset. Subsequent testing resulted in a diagnosis of early Alzheimer's.

At issue for our purposes is that Bob is in many ways uniquely suited as a candidate to participate in this sort of ethnographic work. Bob and Carol rank among the most trusting and sensitive couples I've had the privilege of knowing (ethnographically). Of particular relevance is that they are quite perceptive to subtle changes in their behaviors and are especially self-reflective. For example, prior to his retirement last year, Bob practiced family therapy. His skills and training lend themselves to both self-reflection and to a positive resolution of difficulties. In their own words they are also "…children of the 60's". They are very open and I daresay have a bit of the activist about them in their pursuit of information and care for Bob, working with the local Alzheimer's association, giving TV news interviews and even preparing for congressional testimony in April of 2003.

What's relevant to this work is that Bob's condition highlights important elements of context that appear to be increasingly relevant as his short term cognitive memory degrades (as a function of the disease process). Bob's condition places "context" in relief and permits us to "see" the attributes of context more clearly and more readily than we otherwise might. However, these attributes are not unique to Bob or to disease conditions generally; rather, they are simply more concentrated and observable. Thus, Bob, in this case, is a fair proxy for understanding attributes of context more generally. It's clear all of use rely heavily on "context" in our day-to-day lived experiences, but that it is simply more difficult to observe and delineate. Context is everywhere, if subtle [8]. In Bob's case, context is simply a bit more observable.

2 Methodology

In practice, the use of ethnographic techniques is less of a procedure and more of an exploration; inevitably the real world poses variability requiring one to make on the spot adjustments. One must have a fairly good idea of one's purpose, but how one achieves that purpose may evolve within some constraints.

As mentioned previously, we'd conducted an extensive contextual interview – in their home, on their time, with their artifacts – prior to these two days. Shadowing is a technique by which one acts as "a shadow, ever present, but out of the way. Participant Observation is a technique in which one is present and contributing to the activity and through the contribution, learning about it. Over the two days, depending on the circumstances, I would switch between shadowing and participant observations as appropriate.

I was particularly interested in observing Bob over the course of the two days going about his normal activities – from before breakfast until after dinner, roughly 7:30am through to 8:30 pm. We hesitate to say "perform routine activities" or "conduct his daily tasks" as both linguistically presuppose certain characteristics of linearity, efficiency and effectiveness often not presence in the daily lives of people. Throughout I attempted to maintain a certain distance as appropriate and when I was able. For example, while Bob was preparing to go grocery shopping, I mostly observed (shadowing): taking photos, making notes and occasionally taking video except when Bob offered commentary or asking me to do something that required my participation (e.g., putting something in the car, participant observation). Bob also provided frequent running commentary, which would be a form of a "talk aloud" verbal protocol that he did on his own, albeit for my benefit. I relied on my notebook and on the still image and video features of my Sony DSC 707 digital camera.

One of the symptoms of Alzheimer's is a reduction of short-term memory, i.e., forgetting things. Throughout the two days, I could easily have been Bob's reminder system. However, I wasn't. I did however, act as a "cognitive aid" at certain times characterized primarily by two elements: a) having already witnessed his own repeated failure to remember, e.g., passing the post office for the fourth and last time that day and having not yet mailed the letters in the car's dash, the letters he saw every time he got in the car saying "Oh, I've got the mail those", or b) waiting until the last possible moment before averting a "catastrophic" error, e.g., reminding him to take his gym back as we were pulling out of the garage to go to the gym. In other words, while I tried to not be present, I nevertheless was and therefore had to make accommodations for the research by choosing when and how to interfere.

3 Results

According to David Shenk's, *The Forgetting, Alzheimer's: Portrait of an Epidemic,* it's the little things that begin to fade [9]. Small, inconsequential "lapses", indicate the beginnings of an inevitable decline. Indeed, it's the "little things" that seem to be fading for Bob. He referred to his condition as "progressive incompetence" and the many and varied lapses as *tiny incompetence's*. Little by little, foundational elements of "doing things" randomly begin to flicker on and off and finally don't turn back on. Bob retired almost a year ago, his last year at work fighting the small memory losses that occurred with increasing, but unpredictable frequency. It was a stressful year. Now he's without that stress, but he and Carol must deal with the slowly mounting stress of a partner who is slowly fading away. It brings to mind the Chinese proverb of the frog in a pot of heating water.

Bob and Carol had returned two days previous to my visit from a cruise vacation in the Baja Peninsula. Bob had to get the house going again. They had a list of chores to get through. Over the two days, Bob prepared a grocery list, went shopping, prepared his bag for the gym, went to the gym, bought tickets for a lecture at a local community college, posted some letters, picked up the mail from his mailbox, prepared three meals each of the two days, took two naps, prepared to go skiing but went for a short hike instead, took care of his correspondence and conversed with me about his days and interests. To assist Bob in his daily chores, he maintains a master list of the "things he has to do" for the day. This he keeps in a green stenographer's notebook that almost always sits by his chair or on the dining table.

We will focus the bulk of my comments on his correspondence, or what he called "going through the mail". It is a common enough activity to most of us as to be broadly relevant; but it is also deceptively simple. In fact, it's among the simpler of Bob's activities that we can talk about in this paper, cooking being be far the most complex and, as you will see, from the point of view of a system being contextually aware, "going through the mail" is sufficiently complex to occupy the remaining space of this paper.

3.1 Going through the Mail

One of these little things fading for Bob is his ability to go through his mail, make a decision about each item, and act on it both immediately and at a later time, as warranted by the content of the item itself. In Bob's case, he goes through the stack, makes a decision on what becomes the "last one" and then goes through the stack again, making the same decisions again, then acting on the new "last one". Or, perhaps, in the second time through, he reorganizes the list spatially (one small pile here for emails to write, one pile there for phone calls to make) only to almost immediately forget what the piles mean and to again go through the list. This itself is of interest, but for the purposes of this paper, I'd like to examine what he did to respond to one particular letter, that from "Marsha".

We describe the event here and discuss it in the subsequent section. Bob has already decided (based on a master list of activities for the day) that he needs to call Marsha in response to her very nice card she sent after seeing him on the TV news. That is, he'd already written "Call and thank Marsha for her letter" on his master list. Bob prefers making phone calls to email or writing letters – and says he always has. Bob also believes – always has – in responding to correspondence, especially heartfelt correspondence. Bob worked with Marsha about 20 years ago on a project examining TV violence. On Tuesday (the first of my two days with Bob) Bob and I were in the grocery store and Marsha saw Bob and came up to us. Bob didn't recognize who she was, but after she introduced herself again, he remembered their relationship. In fact, Marsha explicitly thanked Bob for his return phone call. On Wednesday, Bob has "Call Marsha" on his list and proceeds to go through the mail.

1. Bob looks to see what's next on the master list.
2. Call Marsha. He decides to call Marsha, questions who she is and proceeds to go through the mail.
3. Flipping through the mail, he finds a card from Marsha. He re-reads the card: "That's nice."
4. He looks on the card for a phone number. No number.
5. Is there an address? Yes. He decides to get the phone book (from a cabinet in the kitchen not far away). He sits down.
6. He looks back to the card for the name. (See Fig. 1.)
7. Opens the book to "Smith". He realizes there are many Smiths. (It's not really Smith, but it's of similar ilk.)
8. He looks back for the address and it almost seems like he physically gets a hold of the address "19 Kitchen Hill". He decides to search the phone book for the number based on the address because the writer is a woman who's probably listed under her husband's name. He finds Smith: on 19 Tree Hill, 19 Pioneer, 19 Pottery Hill. No luck.
9. New decision, he'll have to write to her instead. He does have the address from the card.
10. He annotates the list and the envelope. He writes a note on the envelope and then highlights the note on the envelope because otherwise "he won't see the note."
11. The envelope is put back into the general mail pile.

As Bob begins in steps 1-5, he sees "Call Marsha" on the master list that he needs to check off. He moves from the list to mail pile. Then he's got at least two things to do: *find* Marsha's letter (in order to) *call* Marsha. As he passes each letter he rejects it as not being from Marsha. He finds the letter and re-reads the card, comments "That's nice". Steps 6-9 comprise a fairly complicated process, but he manages. In steps 10 and 11, Bob annotates the envelope and then replaces the letter in the pile. After realizing the phone number is not to be found, he opts for writing a letter and annotates the envelope as such and furthermore, highlights (with a bright yellow highlighter he keeps nearby) the note he just made because "…otherwise I won't see the note." He then puts the letter right where he won't lose it: in the stack. (Note: Erikson [10] has previously discussed the robust benefits of real space to structure interactions.)

Fig. 1. Step 6 – Bob looking back to the card for the name after getting the phone book.

4 Discussion

Bellotti & Edwards [1] list three relevant characteristics of context aware systems: a) they infer human intent, b) they mediate between people and c) they must be accountable to their users. To achieve these ends, context aware systems must be intelligent, i.e., they must be able to represent what they know and they must enforce accountability based on their inferences. According to Bellotti & Edwards, to be intelligent and accountable, context aware systems must understand context, the most relevant characteristics of which are listed here: identity of relevant people; arrival, presence and departure of relevant people; status and availability of relevant data and actions and their abstractions, such as, "waiting" or "busy"; the purpose and the social rules of the current situation, which in Bob's case might be: correspondence and at leisure.

We rely primarily on Bellotti & Edwards as a theoretical approach for understanding context in this case study because they rely heavily on earlier work and are thus a summary of work to date and moreover, they provide design principles, which we understand as intended to guide design, and which include: a) informing the user of current system capabilities and understandings, b) providing feed-forward and confirmation feedback, c) enforcing identity and action disclosure, such as who is that, who did that, etc., and d) provide control to the user relative to the system.

One possible caveat we mention at the start is that many of their design principles will be familiar to those who design interactive systems, which is, to some extend not the point of a context aware system and moreover not the point of a proactive system [11]. Yet, the principles are sound and they rely on elements of human context in order to act. Thus, these principles, if anything, should be more conservative than what one might offer if one were designing a more extreme proactive system; thus they will suit our purpose here: to examine the meaning of context in within this one relatively simple example.

Let's start with Bob's list. He made the list last evening before bed. By morning, the list had moved to the table where he eats breakfast from the chair in which he sits in the evening. At what point might a context aware system begin assisting/mediating for Bob? Does the system need to know where the list is? Does the system suggest finding the list? If Bob begins taking action on something not on the list, e.g., making breakfast (which was not on the list), should a context aware system note this, mention it to Bob in some way, or perhaps suggest something for Bob to do that is on the list? Does moving the list from the floor near the chair to the table mean he's about to start something on the list? Maybe it is. Maybe it's not. How is a system to know? The value of the notebook's location: on the floor, on the table, may or may not have meaning and it's meaning may be different depending on something known only to Bob. For example, he moved the list off the floor because he's going to vacuum the floor. In the case of an assisting system, suppose that Bob, who has Alzheimer's, moves the list from the chair to the table and then gets "stuck". What was he going to do? Go make breakfast, use the restroom, vacuum or commence with the list? In other words, it's not merely difficult to know where to start – where to start may not be knowable until Bob starts something.

This simple condition – where and when to "start" is a fine example of what we mean by context being **subtle**. It may not even be clear to Bob what his intention is. He may have moved the list intending to start something, then decided to make breakfast instead. If it's not clear to Bob, can it be clear to the system? Which of Bellotti & Edward's elements of context is relevant to the system? To Bob? Perhaps the system should maintain an ongoing dialog with Bob to make that determination. Bellotti & Edwards do not mention that the system may need to work with the user to figure out what it needs to know, though they do suggest the user should exercise control. Should the system inquire whenever it is unsure? What does uncertainty mean for a system in situations of this sort? Do we think Bob wants to be queried by his walls before his morning coffee?

After breakfast, Bob sits and begins to go through his list. Physically, Bob picks up the list and looks at it. He thinks: some items are for later – they are passed over. Some items are for now. Correspondence is now. He decides to give Marsha a call and thank her for her card. One imagines that the system could (and maybe should) provide Marsha's number. But how does the system know that Bob is going to call Marsha: so far, Bob has merely looked a the list. This is another example the subtlety of context. There were about 8 items on the list. Bob decides just by looking. In fact, he may have decided after looking at the last one, not at the item "Call Marsha". In lieu of the system actually knowing the context, what's a poor system to do? Bellotti & Edwards' design principles suggest a solution: The system might have inferred that Bob has decided to do something from the list from his picking up of the list. The system could, it would seem, at this point, reasonably make an inquiry.

A proactive system could provide Marsha's phone number. That much is obvious. However, Bob doesn't want to call her just yet. He doesn't remember the letter, and to make the call means responding to her letter, which he evidently wanted to read again (because he did). Even if the system had indeed inferred correctly that Bob wanted to call Marsha, how could it have known that Bob also wanted to read the card? This is a very good example of how context can be **fluid**. Subtle, seamless shifts from one "contextual milieu" to another suggest a fluidity of context, and yet, based on a logical categorization of activities and stimuli, could easily appear to be quite a different task: going through the mail. That is, a system might infer from Bob's actions that the mail is now the focus of his attention. And it is, albeit in the service of calling Marsha. That is, his reading of the card is not a necessary step in calling Marsha in the abstract; though, for Bob, in this case, it certainly is. (In other examples from Bob's days, context could shift as seamlessly but far more severely, e.g., from cutting potatoes to watching TV because he heard something that interested him; but oh how casually he switched his attention and the context of what he was doing. Should he have been prompted to get back to the potatoes?)

Of course, it's quite unclear how the system could ever know about the letter. Also, who is Marsha? Bob hasn't talked with her in 20 years. Bob perhaps needs to tell the system about Marsha. When did he do this? Did Marsha change her phone number or address? Did she leave a new number in the letter? Bellotti & Edwards [1] discuss clearly the need for the system and the human to have access to information. But for the system to have access to the letter and to link that letter to the list item "Call Marsha" seems a bit improbable.

Even a simple solution can become quite difficult. For example, suppose rather than a green steno notebook, Bob keeps his list on a palm pilot or other hand-held device (e.g., the "activity compass"). Ticking a box near "Call Marsha" would tell the system what the steno book can't. It's easy to imagine. But Bob doesn't like palm pilots. (Bob doesn't carry a cell phone. He doesn't have a beeper. Bob used to carry a portable voice recorder he'd use to remind himself of things; he still uses it as an audible reminder system, but only in the house.) One might assert that using a palm pilot is better than using a steno notebook and is, therefore, the lesser of evils, and Bob should simply acquiesce to the palm pilot. Yet, Bob likes the notebook and at least one reason is that he can easily look back and see all the other items on his prior lists. In fact he stated explicitly that he used to cross off the items until they were illegible; then he switched to a line so he could see what he'd done. Bob is living Bellotti & Edwards' principle of providing status and abstractions of status. And yet, in so doing, a palm pilot would take a good bit of that ability away.

He actually thought about a palm pilot or similar device in his last year of work, but decided against it. Besides, he already had his voice recorder. We must accept that Bob has chosen the steno notebook. We must accept that people do not act in ways others might define as rational. This choice, in addition to his way of going through the mail, is Bob's idiosyncratic way to do things. In fact, there are layers of idiosyncraticity – the notebook, the letter re-reading, his desire to respond with phone calls rather than writing, the list movement, the list placement, the piles of the mail, etc. Thus, the context that surrounds his way of "calling Marsha" is therefore, **idiosyncratic**. Bob's choice of the Green Steno Notebook is but one of many choices he has made over his day, his week, his life. Finally, Bob is again exercising his control of the system by choosing the steno notebook, just as Bellotti & Edwards might suggest. (From the perspective of Alzheimer's. Bob's life strategy of "going through the stack" of mail and making decisions is now failing him; his ability to unlearn a fairly automatic process and relearn a new one is limited – years and years of "going through the mail" has had an effect.)

It sounds trivial, perhaps, to simply assert Bob's choices as an additional challenge for proactive computing in this example. But it is potentially a very real challenge, especially for a man with Alzheimer's. At this point in the disease progression, albeit early as it is, Bob would be hard-pressed to learn how to use a hand-held device. To wit, as we were looking for a frozen stream to hike on Wednesday afternoon, Bob expressed his interest in learning the use the GPS device he has at home. He's highly motivated to learn the device because he simply loves to hike and is loathe to give-up his solo hikes and he knows that a GPS might help him not get lost. But it's "far too complicated" for him to learn, he said, and wishes that someone would spend about 4 days with him to teach him each function, one at a time, as he needs to use it. He's hoping that someone will help him this spring or summer "…before I lose the ability to learn how to use it at all," he said.

Bob was looking for Marsha's letter. He finds it, re-reads it, comments, and then looks for her number as described above. Now, of course, the system, were it to know who Marsha was, could provide the number. In lieu of that, the system could, one supposes initiate a dialog with Bob: "What's Marsha's last name, Bob?" "What's her address?" and proceed to find the number. Not finding the number, Bob decides to write instead, which will be at a later time (**fluid**).

Skipping to step 10, Bob annotates the envelope with a note and then he used a highlighter over that. Bob's compound annotation was written directly on the physical artifact and not, for example, on the master list. For Bob, the task seems linked to the letter rather than to the master list, which is not annotated at all. Moreover, Bob seems to know he needs to be visually stimulated and uses the highlighter to provide himself a contextual cue. (McCarthy [12] suggests that perhaps visual contextual cues might be more relevant to Alzheimer's patients than to others. There may be some validity in this, which requires further work.)

For Bob, the context of his decision is carried by a) his memory, b) the tangible artifact, c) his physical annotation, d) his additional visual stimulus, e) the letter's position with the other letters in the stack and f) the fact that "Call Marsha" is not scratched off the master list. All of these elements are covered by Bellotti & Edward's framework: *status and availability of data*, an *abstraction of status* (will write a letter, which he knows is a secondary decision), *feedback* (*in-process* and *confirmatory*, ironically, which he provides) to himself about his progress to date, *capture of information* relevant to the status of the letter and *subsequent access* of that information (by being in the stack). While Bob's contextual information is theoretically accounted for within Bellotti & Edward's framework, the practical elements – the values – of Bob's context remain *subtle*, *fluid* and *idiosyncratic* and for many intents and purposes remains practically unknowable *a priori*, mostly likely *ad hoc* and perhaps only possibly *post hoc.*

Bob made his list on Tuesday evening. Bob saw Marsha in the store Tuesday afternoon and they talked of the card and his return call. Wouldn't it have been nice if "Call Marsha" would have never made the list that evening.

5 Implications

In theory, there is no difference between practice and theory; in practice, there is. We find this to be an example of this famous quote. While Bellotti & Edwards' [1] theoretical approach – and others (cf., [4], [13], etc.) – seems adequate for encompassing the various categorical elements of context evident in Bob's way of going through the mail, they fail to predict or explain the subtle, fluid and idiosyncratic nature of Bob's "context". Further analyses with other activities may be warranted to more fully examine the frameworks offered.

Theoretically, the abstractions enumerated in the referenced papers account for going through the mail. Yet practically, none of them accounts for the precise details of what context is or how it is maintained. Designing a "proactive" system for cognitive assistance requires accounting for the specifics of context, not just the abstractions. But since the specifics are mostly unknowable, our ethnographic evidence suggests that one simply cannot design such proactive systems in this way. All that this means, however, is that we must shift our research and design emphasis somewhat to accommodate these conditions.

For example, consider Bob's feelings about his green steno notebook. From the perspective of context aware design, one might consider that computational systems must blend with the extant "ways of doing things". That is, the components of the system must not dictate the process, but must, as Weiser [14] suggested "weave" with the long term "way of doing thing" and that it might, over time, work with the person to evolve the system cleverly, perhaps offering new elements of context at just the right moments such that they are useful but so that they don't require new learning. The system remains "invisible".

It's intriguing to postulate that proactive systems, to be effective, must do the learning and that the person with Alzheimer's disease – or indeed anyone - perhaps does no new learning. That is, the person needs to learn nothing. That would be optimal, of course, for Alzheimer's. But what sorts of technologies would be relevant?

A steno notebook comprised of electronic paper might be a good starting point. There needs to be a means of system input that's familiar to the patient and that requires no more management than what they know. A digital notebook that could also sense what's on its list might be terrific. For example, his list could inform Bob in a timely fashion of things to do, for example, remind him to take his gym bag with him when as he's driving out of the garage – not because the system knows when he's going to the gym, but because the bag was packed and the list knows that "go to the gym" is on the list. Actuators strategically placed can coordinate information among the notebook, the gym bag, recent action within the gym bag (it was packed) and a tag in the car that notices the bag is not in the car, but the car is leaving the house and it informs Bob in some pleasant way.

Similarly, in Bob's case again, linking what's on the list – in his writing, with his marks – with activities he's doing would be terrific. For example, his list says "Call Marsha". A digital notebook could "reveal" Marsha's phone number on the list without Bob having to do it. Of course, this takes an enormous amount of coordination. (For more on annotations, see [15]). Another sort of technology would be pens that write with ink, but whose motion transmits what's written to "the system". The system should be able to make certain inferences about what's been written and coordinate with the list and all the other tags and actuators.

In the example from this case study, embedding technology in everyday activities seems to be more about technology becoming, as Weiser said, "invisible", superimposed, or perhaps, under-imposed, on daily activities. Replacing the steno notebook with digital paper is one example. Co-opting the motions of what's otherwise your average pen is another. Adding a few tags to common devices and coordinating their meaning. Embedding technology in such as way as to exercise control, to predict activities, to explicitly enhance and replace cognition in situations that require detailed understanding of context seems rather impossible. To work at the periphery, to coordinate among perhaps simpler, less contextually demanding activities and embed technology appropriately in the situation Gershman, McCarthy & Fano [16] would be of valuable assistance , not only to people with Alzheimer's, but everyone. This is one possible approach for creating supporting technologies. A second approach follows.

In Foucault's [17] master's thesis at Cornell University, she applied to adults an internet search strategy she designed for children. In the children's version, a child enters a word, e.g., tiger. The system returns images of tiger in various contexts. Kids, unable in their elementary years to manipulate language appropriately for Boolean searches can nonetheless identify (though cannot necessarily verbalize) attributes of images relevant to their searches. They then select an image, which is a proxy for verbal attributes we might otherwise enter in a "traditional" search. Perhaps, a similar search technique works for adults, albeit differently. Adults see the images, which initiates a string of associations otherwise not considered. The user can continuously modify the search based on the context of the images without being verbally specific about the context – indeed, without being specific at all. The crucial factor is that not once does the computer know anything about "context" or "intention" as the interaction is entirely under the user's control, and yet can deliver a satisfying experience

For people with early Alzheimer's the difficulty is exactly that the person cannot completely control their interactions with a system. An obvious, but we've argued flawed approach is for a system, as a cognitive prosthesis, to assume increasingly amounts of control over time. Using Foucault as a guidepost, suppose that the system does not assume control of the interaction, but rather acts much more subtly, making unconscious and ambient suggestions to the person. The system could act, less like a drill sergeant and more like a friend making gentle suggestions less precise, but sufficiently accurate perhaps to stimulate memory.

We offer one simplistic example as illustrative of our point, but by no means sufficient. A display occupies the top of the refrigerator. The "system" knows things it knows: time of day, presence of person(s) in the kitchen. That's it. The display is perhaps a washed out image of the kitchen itself. The system can flash images superimposed on the system – a grocery list, a sandwich, a cup of coffee, etc. The point is not to tell the person what to do, but perhaps to jog a memory trace based on a fleeting image, as a possible reminder in the periphery and encouraging the user to perhaps interact with it in some, hopefully subtle way. Smaller displays around the home, e.g., positioned as picture frames, might serve not only to show photos of family and friends, but also as casual, suggestive mechanisms - that can be ignored - and that don't necessarily seek to control the interaction. That is, if we can't know the precise context, can we support in the periphery by providing an enabling environment rather than one that must anticipate and act on the users' behalf?

6 Conclusions

In this paper, we described the deceptively simple example of Bob responding to a mailed greeting card wishing him well. It is, as Tolmie et al. [8] asserts, a rather "unremarkable" routine that "places powerful requirements on any technology that might become embedded in such activities," (p 406). We suggest the further step, however, that proactive computing is less what computing can do for people, that is, on their behalf [11], but rather what computing can *enable people to do*. We focused on what Bob wanted to do - his goals, desires, and needs. Focusing on people, rather than on what technology does, enables us to design appropriate ubiquitous systems that will actually work "unremarkably" in the world [8].

The example was extracted from a larger ethnographic study of Bob, a man diagnosed with early onset Alzheimer's disease. We examined this example in terms of Bob's context as relevant to ubiquitous computing systems. In our examination, we find that the actual values of context, e.g., Bob's notebook location, may or may not vary, but that the meaning of these values themselves can and do vary independently. Further, the values and their meanings can be very subtle to start, and based on something Bob does, their meanings can change almost as if in a flow, i.e., fluidly. We also find that Bob's "way of doing" is just that: Bob's way. The meanings he applies to objects and their attributes and the meanings he extracts, in addition to being subtle and fluid are also idiosyncratic. Finally, we find that these characteristics of context are often expressed simultaneously as Bob goes about living his experiences each day.

Context aware systems that attempt to know what's happening and to guide a person through certain activities in lieu of their own faculties may be overreaching rational boundaries into that of human lived-experience which is more often than not, not as rational as we might like. It may be not only difficult, but also impossible to provide this level of rational support for any activity except for those in the most constrained and proscribed situations. We find advantages to approaching assistance from the periphery by designing systems with that enable human touch-points, rather than trying to reduce them. The examples we've mentioned offer only the briefest of possibilities. There are innumerable ways ubiquitous computing can not only assist people with Alzheimer's disease, but also in which ubiquitous, appropriately context aware computing systems can facilitate the ever increasing complexity of our daily lives.

Acknowledgements

Above all, we must thank Bob and Carol, whose extraordinary candidness and gracious hospitality continues to provide a foundation for inquiry and hope for people with Alzheimer's disease and their families.

References

1. Bellotti, V., Edwards, W. K.: Intelligibility and accountability: human considerations in context aware systems. Human Computer Interaction. 16 (2001) 193-212
2. Chen, H.: An Intelligent Broker Architectuire for Context-Aware Systems. Dissertation. Computer Science. U of Maryland, Baltimore County. (2003) http://users.ebiquity.org/~hchen4/phd/hcthsisp/hcthsisp.html
3. Dey, A.K.: Understanding and Using Context, Personal and Ubiquitous Computing Journal. 5 (2001) 4-7 http://www.cc.gatech.edu/fce/ctk/pubs/PeTe5-1.pdf

4. Dey, A.K., Salber, D. Abowd, G.D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications, anchor article of a special issue on Context-Aware Computing. To appear in the Human-Computer Interaction (HCI) Journal. 16 (2001) http://www.cc.gatech.edu/fce/ctk/pubs/HCIJ16.pdf
5. Dourish, P.: Seeking a Foundation for Context-Aware Computing. Human-Computer Interaction. 16 (2001). http://www.ics.uci.edu/~jpd/
6. Dishman, E., Matthews, J. T., Dunbar-Jacob, J: Everyday health: Technology for adaptive aging. National Research Council on Adaptive Aging. (2003)
7. Malinowski, B.: Argonauts of the Western Pacific. Waveland Press Reprint Edition (1984).
8. Tolmie, P., Pycock, J., Diggins, T., MacLean, A., Karsenty, A.: Unremarkable Computing. CHI 2002 Proceedings. (2002) 399-406 http://portal.acm.org/citation.cfm?doid=503376.503448
9. Shenk, D.: The Forgetting, Alzheimer's: Portrait of an Epidemic, Doubleday, New York. 2001
10. Erickson, T.: From Interface to Interplace: The Spatial Environment as a Medium for Interaction. *Proceedings of the European Conference on Spatial Information Theory.* Heidelberg. (1993) http://www.pliant.org/personal/Tom_Erickson/Interplace.html
11. Tennenhouse, D. Proactive Computing. Communications of the ACM, 43 (2000) http://portal.acm.org/citation.cfm?doid=332833.332837
12. McCarthy, J. Personal communication. (2003)
13. Greenberg, S.: Context as a dynamic construct. Special issue on Context-Aware Computing. Human Computer Interaction Journal. 16 (2001) 257-268 http://hci-journal.com/editorial/vol-16.html
14. Weiser, M.: The computer for the 21st century. Scientific American, 265 (1991) 94-104. http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html
15. Marshall, C.: The Future of Annotation in a Digital (Paper) World). In Harum, Twidale (eds.): Successes and Failures of Digital Libraries. University of Illinois, Urbana-Champaign, 2000, 97-117. http://www.csdl.tamu.edu/~marshall/uiuc-paper-complete.pdf
16. Gershman, A.V., McCarthy, J.F., Fano, A.E.: Situated Computing: Bridging the Gap between Intention and Action, Third International Symposium on Wearable Computing (ISWC '99), 18-19 October 1999, San Francisco. 1999 http://seattleweb.intel-research.net/people/mccarthy/SituatedComputing-ISWC99.PDF
17. Foucault, B. E.: Reintroducing Context in Electronic Search Environments: Comparing Visuals and Text for Contextual Richness. Master's Thesis, Cornell University. (2003)

"Playing with the Bits" User-Configuration of Ubiquitous Domestic Environments

Jan Humble[1], Andy Crabtree[2], Terry Hemmings[2], Karl-Petter Åkesson[1], Boriana Koleva[2], Tom Rodden[2], and Pär Hansson[1]

[1] SICS, Swedish Institute of Computer Science, Box 1623, SE-164 29 Kista, Sweden
{humble,par,kalle}@sics.se

[2] MRL Lab, Jubilee Campus, University of Nottingham, Nottingham, NG8 1BB, UK
{axc,tah,bnk,tar}@cs.nott.ac.uk

Abstract. This paper presents the development of a user-oriented framework to support the user reconfiguration of ubiquitous domestic environments. We present a lightweight component model that allows a range of devices to be readily interconnected and an editor to support users in doing this. The editor discovers available ubiquitous components and presents these to users as jigsaw pieces that can be dynamically recombined. The developed editor allows users to assemble lightweight sensors, devices such as displays and larger applications in order to meet their particular needs.

1 Introduction

The emergence of increasingly interactive ubiquitous environments makes it important that we consider how devices are placed within an environment, how combinations of these devices are practically managed, and how these devices work as an ensemble. This is particularly true in domestic settings where inhabitants have considerably more control over the environment than in office or work settings. An essential element of domestic environments is their evolutionary nature. Domestic environments are open to continual change and the need to support this change is essential to the successful uptake of ubiquitous devices in domestic spaces. Previous studies have highlighted how inhabitants continually reconfigure domestic spaces and the technologies within them to meet particular demands [7]. Similarly, architectural historians, such as Brand [2], have highlighted the importance of change to allow inhabitants to appropriate and adapt domestic spaces to meet their evolving needs. The dynamics of change highlighted by Brand are particularly significant when we consider the nature of ubiquitous computing in domestic environments [8]. As Edwards and Grinter [4] point out, unlike the lab houses that largely provide the focus of current research in the area, the networked home of the future will not be custom designed from the start but will emerge in a piecemeal fashion. The evolutionary development and the piecemeal introduction of ubiquitous computing into the home

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 256–263, 2003.

environment demand that we consider how real world processes of adoption and use might be supported to permit the ongoing configuration and reconfiguration of interactive devices.

2 Configuring Devices within the Home

In developing technology for domestic settings we need to make sure that it is open to continual change. Inhabitants must be able to quickly place devices in the home, understand this placement and rapidly reconfigure device arrangements. If the cost of change is too high then devices are less likely to be added to the home and become part and parcel of the 'everyday stuff' that is routinely used and exploited by its inhabitants [10]. A central issue in the development of innovative ubiquitous devices for the home is the relationship between devices and the infrastructure they exploit. In fact, as interactive devices become increasingly ubiquitous the underlying infrastructure supporting them needs to become increasingly prominent and available to users.

While infrastructures such as Jini[1], UpnP[2], and the Cooltown[3] reduce the cost of introducing new devices they are oriented towards the *developer* of new devices rather than the eventual *users* of these devices within a ubiquitous environment. The focus of these infrastructures has by necessity been on the development of appropriate protocols and techniques to allow devices to discover each other and make use of the various facilities they offer. Limited consideration has been given to how users may be involved within the dynamic configuration of these components. Two notable examples are the Speakeasy system [6], which has adopted a composition model based on typed data streams and services, and iStuff [2], which knits together a number of ubiquitous devices via a state based event-heap. Our composition model differs from that of Speakeasy and is closer to iStuff in that we seek to exploit a distributed shared state model. However, our challenge is to allow users to understand the arrangement of devices and manipulate to these in order to meet their changing household demands.

2.1 A User Oriented Component Model

Our user-oriented model exploits a component model based on JavaBeans[4] and shares bean properties across a distributed dataspace. Component based models seek to balance between the benefits gained from assembling existing components and the loss of expression and generalization. The majority of component models have focused on software components with an emphasis on supporting the programmer. In our case we wish to develop a component model that embraces a heterogeneous col-

---

[1] Jini - http://wwws.sun.com/software/jini/

[2] Universal Plug and Play - http://www.upnp.org

[3] Cooltown - http://cooltown.hp.com/cooltownhome/

[4] Java Beans - http://java.sun.com/products/javabeans/

lection of devices and is readily understood by inhabitants of the home. Consequently we have tended to focus on making the composition as simple as possible despite the reduction of expression.

The basis of our component model is a shared dataspace that allows real world devices to make information about the nature of the physical environment digitally available. Devices can use this dataspace to become aware of their context and represent this contextual information to other devices and to make this manifest in the physical world. The aim of devices within the physical environment is either to make information from the physical available within the digital or to make digital information have a corresponding physical manifestation.

The fundamental aim of components is to ensure the convergence of the physical and the digital environment. Thus each component can be though of as a **digital/physical transformer**. There are three main classes of transformer component.

- *Physical to Digital Transformers* take physical effects and transform them into digital effects. Each transformer measures a physical effect and transforms it into a corresponding digital property that is shared through the dataspace.
- *Digital to Physical Transformers* represent the complementary set of transformers. Their job is to make digital information physically manifest by transforming the values of shared properties to drive some sort of physical device.
- *Digital Transformers* act upon digital information and effect digital information. This class of components provides a way to present to users deeper semantic reactions to changes in the environment.

This model is analogous to the one proposed within iStuff which provides developers with a set of discrete devices that can be assembled through publication within the event-heap. In this paper we build upon this work by considering how components such as the devices in iStuff and the ways in which they are configured might be exposed to users. Consequently, our emphasis is on the development of user-oriented editors that allow the dynamic composition and assembly of arrangements of devices. Below we focus on the design and development of one users editor – the jigsaw editor -that we have developed in partnership with users through a series of focused cooperative development exercises.

3 From Transformers to Jigsaw Pieces

Our starting point for the development of a user-oriented approach to reconfiguration was the formulation of a simple editing metaphor based on the notion of assembling simple jigsaw like pieces. Our choice of the ‘jigsaw pieces’ metaphor is based on the familiarity evoked by the notion and the intuitive suggestion of assembly by connecting pieces together. Essentially, we allow users to connect components and compose various arrangements through a series of left-to-right couplings of pieces. Constraining connections in a left to right fashion also provides users with the sense of a pipeline of information flow.

Our development of the jigsaw model exploited a paper-based 'mock up' approach [5] married to 'situated evaluation' [11]. In order to structure our investigation we presented users with a set of initial seed scenarios elaborating various transformers and their potential arrangement. These reflect different levels of abstraction and provide a *starting point allowing users to reason about* the editor, the complexity of configuration, and the nature of ubiquitous computing in the context of their everyday lives. The seed scenarios were drawn from earlier ethnographic studies [3], and some initial prototype development. An illustrative selection of scenarios and components are presented below.

Seed Scenario #1. A Common grocery item is missing from a kitchen cupboard

Using the pieces shown below, **GroceryAlarm** is connected to **AddToList,** which is then connected to **SMSSend**. **GroceryAlarm** reports the missing item after a certain time interval and the missing item is added to an electronic shopping list. This list is periodically sent via SMS to a mobile phone.

GroceryAlarm: Generates names of missing groceries in the cupboard. It detects groceries moving in and out and if one is away more than 30 minutes it is said to be out.

AddToList: Takes an element string and adds it to the list it publishes into the dataspace.

SMSSend: Takes a message string and sends this as SMS to the given phone.

Seed Scenario #2. Reminders can be directed to a number of outputs

A reminder application lets the user enter textual or auditory reminders using a touch display and microphone. This can be connected to a display, sent to a mobile, etc. The **Reminder** piece can be connected to the either the **KitchenTableDisplay** or **SMSSend** to allow users to receive reminders where they are most appropriate.

Reminder: Corresponds to a reminder application that provides an input GUI, manages the reminder alarms, and publishes reminders as URLs when reminders are due.

KitchenTable Display: Takes a URL and displays the associated web page.

SMSSend: Takes a message string and sends this as SMS to the given phone.

Our mock up sessions suggest that users felt a need to understand how the devices within their home where interconnected but few of them expressed any desire to "program their home". Rather users expressed the need for a simple access point that allowed them to both view the assembly of devices and to reconfigure this assembly as they used these devices for different purposes. Responding to this request we constructed the Jigsaw Editor Tablet described briefly in the following section.

4 The Jigsaw Editor

The editor presents the transformers within the domestic environment as Jigsaw pieces on a tablet based display. The editor discovers a local dataspace where transformers are registered. In order to make itself available for use a transformer exports itself to the distributed dataspace. The transformer is introspected and the properties associated with it are made available as input and output points, each transformer also has a jigsaw piece property which exports how it should appear in the editor. Each room has a dataspace associated with it and components that can be accessed from that room are registered with the dataspace.

It is worth stressing that within this approach we are constraining the potential for expression. Our emphasis is on reconfiguration rather then programming and we do not seek the richness of programming expression allowed by iCap [9] or existing visual programming languages which emphasize the development of new applications and the understandably of procedural programming constructs. Rather we wish to allow users to understand their environment and how the devices within it are interconnected and to allow them to change these interconnections through an editor.

The Jigsaw editor is made available to users using a tablet PC that uses 802.11 to talk to the dataspace (see Fig. 1). The editor discovers the dataspace and is notified of the components available within the dataspace. The editor is composed of two distinct panels, a list of available components (shown as jigsaw pieces) and an editing canvas. Jigsaw pieces can be dragged and dropped into the editing canvas or workspace. When a jigsaw piece is dragged onto the workspace it clones itself and becomes a symbolic link to the underlying component it represents. The editing canvas serves as the work area for connecting pieces together and visualizing their activities.

Connecting jigsaw pieces together works by dragging a particular piece in the vicinity of a fitting target piece. When a jigsaw piece is first dragged, non-compatible pieces in the workspace are disabled and shadowed, indicating which pieces are plausible target connections. When a user places two components close to each other they snap together if the property types for the underlying components input and outputs match. Audible feedback is provided when they snap together. When a connection is made a link is registered with the dataspace and the property of the source component is connected to the compatible property of the target (or sink) component. This link is persistent and the assembly is displayed whenever the editor reconnects. Components have the option of containing several properties making multiple connections possible. For one-to-one matching the connection is done automatically. For multiple choices a dialog window is presented to the user asking for the preferred matching for each property.

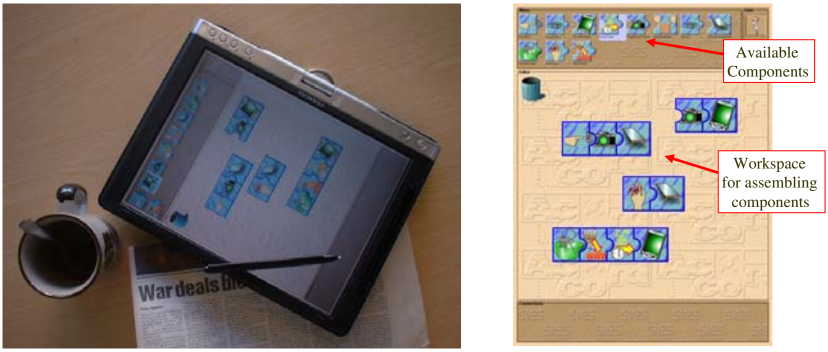


Fig. 1. The tablet editor and the editor screen

The editor not only provides audio feedback on interaction, but also traffic feedback. When properties related to jigsaw pieces in the dataspace are updated, the corresponding jigsaw piece changes colour and a short audio clip is played. Users can monitor the connections between components in the home using these indications of activity. The editor allows simple sensors such as switches, more complex devices such as displays and on-line applications to be interconnected. In the rest of this section we illustrate some of the assemblies developed by users during our evaluation sessions..

4.1 Exploiting a Simple Sensor: The Doorbell

Responding to requests by users for a doorbell sensor we extended the set of components to provide a simple touch sensitive component. This component utilizes the Smart-Its toolkit[5], a general-purpose hardware toolkit for ubiquitous devices. A Smart-Its device has a board equipped with sensors and attenuators and a communication board supporting wireless connection to a base-station. A multi-device event engine has been developed that maps readings from the sensors into Java events which can then be used to changes properties in a JavaBean that acts as a proxy for the sensor device. This arrangement is show in Figure 2, which also shows the Smart-It device and the corresponding jigsaw piece made available to the editor.

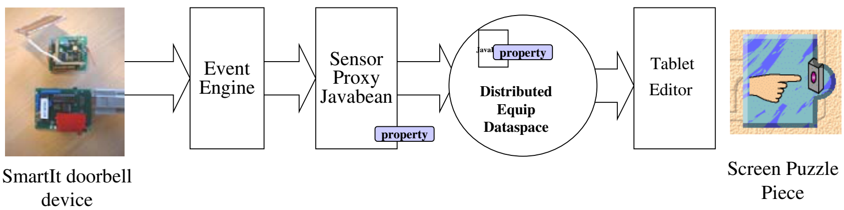


Fig. 2. Making simple lightweight sensors available to the dataspace

[5] Smart-Its - http://smart-its.teco.edu.

Once made available to the dataspace the Smart-Its device appears on the jigsaw editor and users can connect the sensor device to other components. For example, the sensor can be used to drive larger scale devices connected to the dataspace. Two such devices are the web camera and a portable display that is build using an iPAQ (a similar component exists for MMS messages to mobile phones).

4.2 Integrating Larger Devices: The Webcam and Display

Larger devices are made available to the system in a similar way as lightweight sensors and actuators. Essentially the driver used to connect to the device is 'wrapped' as a JavaBean. This JavaBean is then exported to the dataspace with the result that the corresponding property values can be shared across the dataspace. This means that the device is available to the jigsaw editor and can be combined with the inputs provided by the lightweight sensors. For example, the arrangement shown in Fig. 4 shows the pushbutton being used to signal a webcam to take a picture. The picture taken by the webcam is available as an output. Linking the webcam piece to a portable display means that this picture is then directed to that display. The arrangement as it appears on the editor screen is shown below.

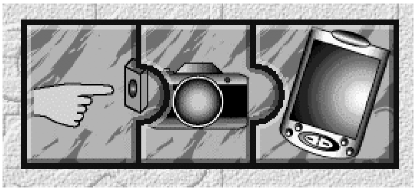


Fig. 3. Combining the doorbell the webcam and the portable display

4.3 Exploiting Applications

Users also sought to link sensors and devices to more abstract and complex entities. We address this issue by exploiting the ability of JavaBeans to make applications into components. By exporting the properties from the JavaBean used to represent an applications we can it available to users and combine the application with lightweight sensors and devices.

For example, to support remote monitoring users can connect a lightweight Smart-It motion sensor to a more complex webcam device and make the output from the device available to a weblog application. This configuration means that whenever motion is detected within a space this is used to take a picture that is then automatically added to the weblog. Users away from home can access the weblog to view the image via any web browsers..

5 Conclusions

In this paper we have presented the development of an editor that supports user-configuration of ubiquitous computing environments in the home. The editor exploits a jigsaw metaphor to make user-configuration of complex functions readily intelligible. It is worth reflecting on the heterogeneity of the components users wished to connect together. It was not unusual to see users develop assemblies that combined lightweight sensors with more traditional computer devices and larger applications and services. In order to allow user to construct these heterogeneous assemblies of devices we developed an editor that discovers components that have made properties available to a distributed dataspace. This arrangement allows quite distinct types of component to offer a very simple lightweight state based interface, which the editor can then, present to users to allow them to construct assemblies to meet there particular needs.

Obviously, the process of user-based development is ongoing, with each iteration leading to the further refinement of the technical infrastructure and toolkit of devices, software and applications that embed ubiquitous computing in the domestic environment to meet real user needs. We are currently in the process of placing the toolkit in a number of users' domestic environments for continued assessment. We envisage these trials raising significant issues of access control and management.

The current version of the toolkit including the Jigsaw editor is publicly available and may be downloaded from *http://www.sics.se/accord/toolkit.html.* This allows developers to wrap their particular sensor, devices or applications as JavaBeans, to provide an iconic representation of the device and to publish them to our dataspace. Once within the dataspace they become available for use through the Jigsaw editor. Our aim is to allow users more control over the assembly of the ubiquitous devices that share their environment in order that users within the home can readily situate ubiquitous technologies in the space they live in.

References

1. Ballagas et al. (2003) "iStuff", *Proc. CHI '03*, Florida: ACM Press.
2. Brand, S. (1994) *How Buildings Learn*, New York: Viking.
3. Crabtree, A., Hemmings, T. and Rodden, T. (2002) "Pattern-based support for interactive design in domestic settings", *Proc. DIS '02*, London: ACM Press.
4. Edwards, K. and Grinter, R. (2001) "At home with ubiquitous computing" *Proc. Ubicomp '01*, pp. 256-272, Atlanta, Georgia: Springer-Verlag.
5. Ehn, P. and Kyng, M. (1991) "Cardboard computers", *Design at Work* (eds. Greenbaum, J, and Kyng, M.), pp. 169-195, New Jersey: Lawrence Erlbaum.
6. Newman, M. et al. (2001) "Designing for serendipity", *Proc. DIS '02*, London: ACM Press.
7. O'Brien, J. et al. (1999) "At home with the technology", *ACM Trans. on Computer-Human Interaction*, vol. 6 (3), pp. 282-308.
8. Rodden, T. and Benford, S. (2003) "The evolution of buildings and implications for the design of ubiquitous domestic environments", *Proc. CHI '03*, Florida: ACM Press.
9. Sohn, T. and Dey, A. (2003) "iCAP", *Proc. CHI '03 (interactive poster)*, Florida: ACM Press.
10. Tolmie, P., et al. (2002) "Unremarkable computing", *Proc. CHI '02*, pp. 399-406, Minneapolis: ACM Press.
11. Twidale, M., Randall, D. and Bentley, R. (1994) "Situated evaluation for cooperative systems", *Proc. CSCW* '94, pp. 441-452, North Carolina: ACM Press.

IntelliBadge™: Towards Providing Location-Aware Value-Added Services at Academic Conferences

Donna Cox, Volodymyr Kindratenko*, and David Pointer

National Center for Supercomputing Applications (NCSA)
University of Illinois at Urbana-Champaign (UIUC)
605 E. Springfield Ave, Champaign, IL 61820, USA
{cox,kindr,pointer}@ncsa.uiuc.edu

Abstract. This paper contains details on a project aimed to provide location-aware value-added services to the participants of an academic conference. The major characteristic of this project is the fusion of RFID technology, database management, data mining, real-time information visualization, and interactive web application technologies into an operational integrated system deployed at a major public conference. The developed system tracks conference attendees, analyzes the tracking data in real-time and provides various services to the attendees, such as a real-time snapshot of the conference events attendance, the ability to locate friends in the convention center, and the ability to search for events of interest. The results of this experiment were revealing in terms of both the potential of the developed technology and the conference dynamics.

1 Introduction

Every year thousands of conferences and professional trade shows take place worldwide, attracting millions of attendees. These conferences represent important venues for social interaction and knowledge exchange, providing a place to find others who share common or complementary interests. However, the difficulties of event management and missed opportunities for communication at conferences continue to be vexing challenges for organizers. How do attendees find others who share their interests? How do they identify and communicate with others whose expertise they seek? How can organizers understand the dynamic profile of participants at the sessions and be more responsive to their emerging interests? In the absence of good answers to these questions, many attendees do not benefit from the events as much as they could.

The IntelliBadge™ project is an academic experiment that uses smart technology to track participants at public events and provide them with value-added, personalized, location-aware services with the goals to facilitate social interactions and foster social networks among the conference attendees. IntelliBadge™ was first publicly showcased at IEEE Supercomputing 2002 (SC2002), the world's premier supercomputing conference, in the Baltimore Convention Center, November 16-21, 2002. The SC2002 organizing committee in part funded this project to push the technological envelope at the conference, to provide a fun and value-added experience for technical

* To whom correspondence should be addressed.

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 264–280, 2003.

program attendees, and to collect data that could provide important, useful information to the organizers.

While the SC2002 committee had basic requirements for the project as stated above, we identified 3 major system design concepts that, along with the budget constraints, served as the main driving force during the project implementation phase. First, we wanted to reliably track volunteer participants in real-time during the conference. Ease of use for the participants and the transparency of the underlying technology were concomitant goals. The convenience and security of the IntelliBadge™ registration drove the design of the process and software. We preferred to get a richer database, but decided it was more important to make the process user friendly.

Second, we wanted to enable attendees to conveniently access services within a secure environment. We decided to build a web application to provide convenient access to services via the web. The most important services included finding people and events; the goal to provide participants with useful information drove these decisions. We also decided to deploy web kiosks for an on-site access to the web application.

Third, we wanted to provide real-time visualization displays showing conference data. We determined that no personal information would be shown on these public displays; rather we would visualize general aggregate data. We decided to provide multiple large-scale visualization displays for attendees to view. The overall visual design goals were to create a project identity and visual unity across a diverse set of applications; to provide variety of interactive visualizations to engage and interest attendees; to communicate the project to as many people as possible using available imaging technologies.

2 Related Work

In recent years, attempts have been made to provide conference attendees with various value-added services based on the ability to either track individual attendees as they go from one location to another or detect when they interact with each other. Thus, Meme Tags project [1], used electronic name tags capable of exchanging short messages (memes) via infrared (IR). The tags also stored information about the interaction between tag wearers and shared it with the centralized database. The cumulative data was shown on large displays (Community Mirrors). The overall goal of the project was to develop a technology that would allow to study community formation process, social networks, and how ideas propagate in the community.

Digital Assistant developed at ATR Media Information Science Lab [2] aimed to enhance communication among conference participants by tracking them as they attend various locations and providing access to a content-rich personalized environment either via web kiosks or interactive displays. Users were required to wear IR badges that could be detected at some locations within the conference space. The resulting data was used to create the user's touring diary and to provide personalized real-time services.

Georgia Tech's Social Net system [3] required attendees to carry a portable radio frequency (RF) device, called Cybiko, to help mutual friends connect strangers (who were co-located for a considerable amount of time). In order for these mutual friends to identify who among their friends are not connected (but should be, because they tend to be co-located), the system requires each user to provide a list of all their friends – a task that turned out to be challenging for some in a field test of 10 users at a 3-day conference.

nTag by nTAG Interactive, LLC (www.ntag.com) uses semi-passive Radio Frequency IDentification (RFID) tag operating in the UHF band which enables a conference organizer to use it for security, to record how many people attended certain sessions, or to track how many people visited certain areas of an exhibition floor. When people meet, their tags exchange information about their interests and preferences. Tags also store and provide convenient access to the conference program.

CharmBadge by Charmed Technology, Inc. (www.charmed.com) uses IR-based tags programmed with attendees' individual business card information. This information is exchanged between attendees as they interact with each other and the interaction is logged and subsequently uploaded to a private website accessible by each user. The system does not provide data to the users in real-time and does not track user location in real-time.

SpotMe system by Shockfish SA (www.spotme.ch) requires participants to carry a cell phone-size device via which they can find out who is standing within a 30 meter radius from them. Participants can be notified if a person with shared interests comes within 10 meters, and they can send messages to each other or exchange electronic business cards. SpotMe does not provide services based on the knowledge of who attended what events since there are no people tracking capabilities built into the system.

IntelliBadge™ differs from the above-described systems both in terms of the core technology used to track people and in terms of the end-user applications. It implements *location tracking by proximity* to RF location markers installed at the points of interest. All the user services are built around tracked location information and a priori knowledge about the attendees and the conference events. IntelliBadge™ is an integration of RFID, database, web technology, and data visualization into a production-quality system.

3 Project Overview

IEEE Supercomputing conference is an annual event organized by a group of volunteers with the sponsorship of IEEE Computer Society and ACM. It consist of a number of simultaneous events, such as Tutorials, Technical Program, Educational Program, Exhibits, etc. It is typically attended by over 5000 attendees; usually about 2000 of them register for the full conference. Since the full conference registrants typically tend to stay throughout the entire conference and participate in a variety of events, we decided to tag and track only this group of people because in our opinion they would most likely benefit from the type of services that we were providing.

SC2002 conference attracted 7,240 participants with 2,188 paid technical program registrants eligible to participate in the IntelliBadge™ project. IntelliBadge™ participants were required to carry a small active RFID tag in a small plastic envelope next to their regular conference tag. During the registration, they were given the tag and were directed to a registration kiosk (Fig. 1). At the kiosk, they were required to scan the tag by the barcode reader (RFID tags had barcode labels with the same unique ID as RFID) and to create user name and password for later access to the kiosks or the web server. Once the login is set up, users were asked to fill in a short personal profile either by scanning 2D barcode from their regular conference tag, or by typing it manually. Users also had an option to create their own group or join one of the groups created by their friends. In addition, they were also asked to indicate their interest level in 10 conference-related subjects. In total, 890 conference technical program attendees registered to participate in the IntelliBadge™ project. At the end of the conference, participants were required to return the tags. In total, 752 tags were returned.

Fig. 1. IntelliBadge™ registration booth and one of the IntelliBadge™ displays.

Once registered, participants could access kiosk services either via one of 8 on-site kiosks (used either as registration kiosks or as service kiosks depending on the tag scanned by the kiosk's barcode scanner) or via their own laptops on-site or off-site. They could modify their registration profiles, use various IntelliBadge™ services, or read about the project. They could also stop by one of 3 on-site large displays (Fig. 1) and either passively observe the conference activities as tracked by RFID tags, or interact with the display applications. We encouraged registered participants to use the provided services as much as possible by giving away some prizes. Thus, at the end of the first day of the conference, we awarded 10 of those participants who spent most the time standing in front of the large displays. At the end of the second day, we awarded prizes to 10 most frequent users of the kiosks. At the end of the third day we awarded prizes for those participants who walked the most distance (as tracked by RFID tags) during the 3 days of the conference. Finally, at the end of the conference we awarded a grand prize to a randomly selected participant.

Privacy and safety concerns were of a great importance to us, therefore we opted to accommodate voluntary rather than mandatory participation. Although providing any registration information was not required, we encouraged participants to provide as much as they would feel comfortable since the quality of the value-added services heavily depended on the availability of the data about the participants. Any portion of the information that users provided us could be marked as private or public, thus allowing a finer level of user control about what to share with others on-site. This, however, had no impact on the applications where aggregate data was presented. We also disclosed to the users our intention regarding the post-conference usage of the data.

4 Hardware Architecture

Basic hardware architecture of the IntelliBadge™ system is shown in Fig. 2. The Savi Technology equipment (described below) forms the front end of the data collection system. The remaining blocks whose names are prefixed by "ib" are separate networked computers that provide downstream data gathering, processing, display, and user entry.

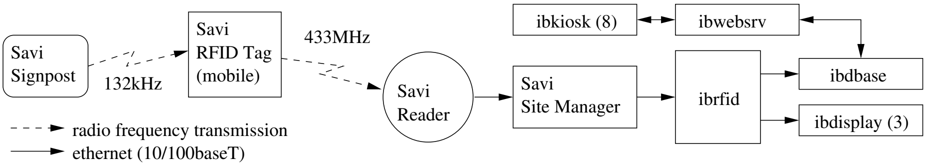


Fig. 2. Basic IntelliBadge™ hardware architecture.

The **ibrfid** server is responsible for receiving event messages from the site manager and adding each message to a local message queue. Various daemons read the message queue, check the messages against some defined criterion and act accordingly, e.g., send an event over the network to the three display servers (**ibdisplay**) via a defined UDP port, or store data in the database, **ibdbase**. The three display servers, **ibdisplay**, run custom interactive visualization applications that show current and historical information collected by the IntelliBadge™ system. There is also an interactive component to the display servers in that they would display a person's organization name on a map of the world. The database server, **ibdbase**, was used for data repository. The web server, **ibwebsrv**, provided the user web browser client screens at the kiosks and also forwarded the user data input at the kiosks to the database.

The IntelliBadge™ kiosk machines (**ibkiosk**) run Microsoft Internet Explorer browser in kiosk mode. They also have a barcode scanner attached that scans 1D barcodes from RFID badges and 2D barcodes from conference cards and sends URL requests to the browser running at the kiosk with the scanned data. Each RFID badge has a 7-digit identifier which the application uses to identify the badge. The conference cards have user information such as name, address, company, etc.

The Savi Technology device that is actually tracked is the RFID tag, a small and lightweight device typically carried in a front shirt pocket or in a conference badge holder. The tag has two concurrent operating modes: beacon, which may be disabled, and signpost detection, which is always enabled. In the beacon mode, the tag sends its unique tag id and a beacon event to the Savi Reader once each programmed beaconing interval. In the signpost detection mode, when a tag enters a Savi Signpost's range (programmable from roughly one to eight feet), the tag receives a unique signpost id number and transmits this id, a *detect* event, and its own tag id to the Savi Reader. When the tag leaves a signpost's detection range, the tag transmits the signpost id, an *undetect* event, and its tag id to the Savi Reader.

The Savi Technology Signpost can be thought of as simply an active location marker. It transmits its unique signpost id on a 132 kilohertz carrier at some programmable time interval. The signpost may also be used to program tags that come within range, e.g., turn beaconing on or off. The communication between the tag is unidirectional from signpost to tag.

The Savi Technology Reader is a 10baseT network-connected device that listens for tag communication with a 433.92 megahertz receiver. As with the signpost and tag, the communication between the tag and reader is unidirectional, from the tag to the reader. The effective reception range of a reader is roughly 300 feet. The reader takes the information received from the tag, adds a timestamp, and communicates this information to a Savi Site Manager via the proprietary Savi Universal Data Appliance Protocol (UDAP).

The Savi Site Manager is an embedded-NT computer that understands UDAP and provides a central collection point for some arbitrary number of readers on the network. The usual Savi Technology system data flow continues on into more of their asset tracking system, but the IntelliBadge™ adaptation of the Savi equipment breaks with the Savi data flow at this point. Instead, the IntelliBadge™ **ibrfid** server listens for network connections from the site manager on a defined TCP port. These network connections contain event messages that include all of the information sent to the site manager by the reader(s), with the addition of the IP address of the reader that sent the event message.

Tracking coverage in the Baltimore Convention Center was provided for 4 rooms on 3rd floor used for the Conference Technical Program events, Ballroom on 4th floor, and Exhibit Hall on 1st floor of the building. The Ballroom was tracked with a single Savi reader located above one of the entry doors. The attendees' tags would beacon every minute and would be picked up by the Ballroom reader. Entry doors to 4 rooms in the Technical program area were equipped with 4 signposts each: 2 inside the room and 2 outside the room. This allowed to reliably detect attendees as they enter or exit each room. In addition, 2 Savi readers were installed above the entry doors to two of the rooms. This was necessary to detect tag events from the tags located in the Conference Technical Program area. This setup allowed to track attendees as they go in and out of the rooms and also to track if they are present near the rooms. The data collected on 300 level was sufficient to identify who/how long was present in each conference room and on the floor in general. The Exhibit Hall was tracked with four Savi readers. The attendees' tags would beacon every minute and would be picked up by one of these readers.

All 3 IntelliBadge™ displays were equipped with Savi signposts to provide tag detection for real-time interactive applications. The display at the entrance to the exhibit hall was also equipped with RFID reader since readers in the exhibit hall were located too far away to provide adequate coverage for the location of the display.

5 Software Architecture

The architecture of the software developed to support IntelliBadge™ SC2002 demonstration is shown in Fig. 3. All of the software on the Savi Technology equipment is proprietary, and so all of the front-end data gathering devices are treated like a black box for the purposes of this paper. The *tagd* daemon running on **ibrfid** provides the interface to the Savi Technology black box. It receives event messages from the Savi Site Manager, extracts information from the message, and writes data to a local event message FIFO queue. The User History Daemon, *uhd*, uses event messages from the queue and previously processed event messages retrieved from **ibdbase** to keep user historical data (current user location, location history, and mileage) current in **ibdbase**. The daemon *app_disp* keeps a static table of display servers that need to be notified if a given event message criterion are met. It gets the most recent unread message from the event queue, checks the message against the defined criterion, and passes the event on to the defined display server(s). In this way, a display server is notified if a badge was detected near that display server's signpost.

The server **ibdbase** provides the user IntelliBadge™ database services using mySQL-MAX database software. These include data related to user location history as well as data input or modified by the user at the **ibkiosk** terminals. The server **ibwebserv** provides the user interface to an IntelliBadge™ user's Internet browser. IntelliBadge™ website was implemented in JSP.

The main visualization application, *vmain*, is an OpenGL-based framework that allows switching between several visualization schemas. Each such schema is implemented as a display callback that is called by the main application. As a result, the main application maintains the OpenGL context, handles events, and runs the main execution loop. The switching between various schemas (*vs1*, *vs2*, and *vs3*) is done based on the time of day. Multiple instances of the application can therefore be synchronized by synchronizing the system clock on each machine via ntpd.

The daemon *geoeventd* running on **ibdisplay(s)** provides real-time notification for the *vmain* display program when a tag approaches one of the displays. When a *detect* event message is received, *geoeventd* translates the detect event message into a Maya Extension Language (MEL) script command and feeds this to *vmain* application which uses VMaya™ library routines to interpret the script. *Vmain* centers the tag bearer's institution location on the map of the Earth by modifying the parameters of a rigid-body dynamics simulation that drives the current camera position, and then displays that institution's name and a flag to mark the spot. Similarly, when a tag bearer leaves the area of the signpost near the display, an *undetect* event message is received by *geoeventd*, and *vmain* removes the institution's name and marker flag from the global map.

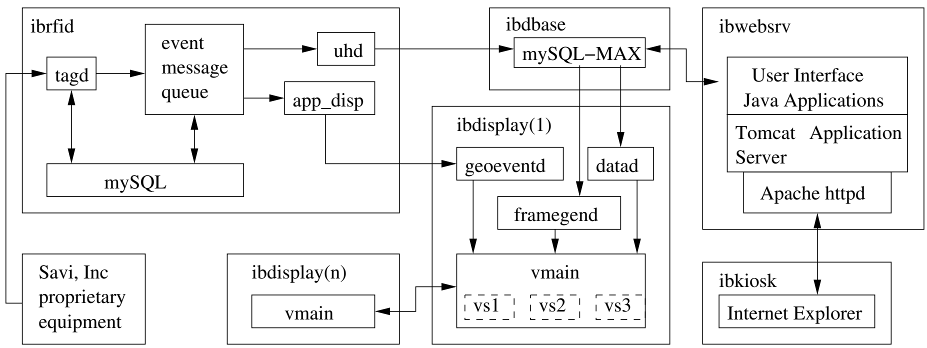


Fig. 3. IntelliBadge™ software architecture.

The daemon *datad* utilizes the database on **ibdbase** to replay the last three hours of data visualization on the display application. The daemon *framegend* utilizes the database on **ibdbase** to generate new frames for "how does your conference grow" visualization schema (see next section for details) using Maya 3D animation software.

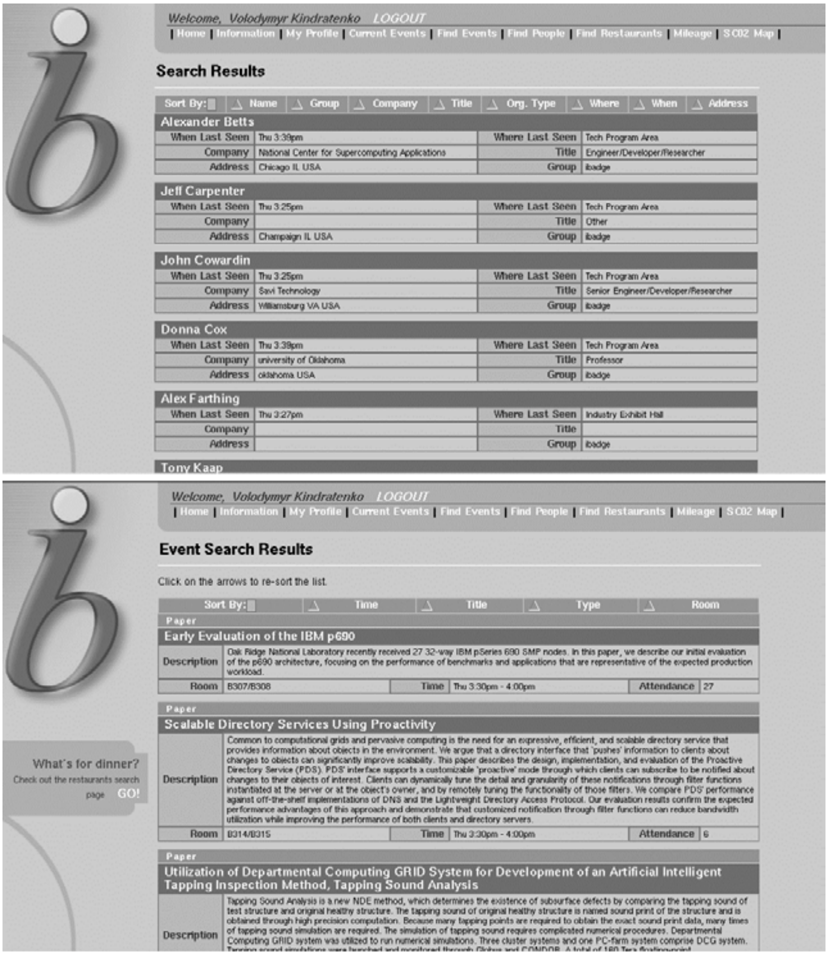
VMaya™ is a C++ library for real-time rendering of scenes created with Alias/ Wavefront's Maya 3D animation software. The library traverses a Maya scene using Maya's C++ API, and constructs a representation of the objects. It observes changes to their shape and position, and renders the results using OpenGL. VMaya™ provides a way to connect real-time data with the type of imagery produced by Maya to create dramatic, narrative visualizations.

Media Analysis Learning and Transport Infrastructure (Malti™) was deployed to enable streaming video between different displays Malti™ is a multi-layered communications infrastructure with real-time multi-channel media processing and transport capabilities. The transport layer abstracts RTP, UDP and TCP transport mechanisms to simplify application integration. The application layer allows multi-channel media processing applications to be built by linking together modules into graph structures. Both VMaya™ and Malti™ were developed in-house.

6 Services and Applications

We provided IntelliBadge™ participants with 2 types of applications: web kiosk and display. Access to the web kiosk application was provided via kiosks (Fig. 1) located in the Technical Program area and in the Exhibit Hall and via the web. Large rear-projected displays were installed in the Technical Program area, at the entrance to the Exhibit Hall (Fig. 1), and in the Exhibit Hall.

The web application was designed as a live conference web site with an emphasis on presenting results of RFID badge tracking and conference events. When accessed from the kiosk machines, it also acts as a user registration interface that allows to create logins and associate user information with the RFID badge using the scanner as a data input mechanism. Hence, the web application has to vary the presentation, depending on whether it is being run at the kiosk or not, and whether the user has logged in or not. After login in, users were directed to the index page with links to various IntelliBadge™ services. The main services were Conference Attendees Search and Conference Events Search. Attendees Search service allowed to locate individuals or groups of people based on such criteria as person's name, institution, interests, address, etc. These search criteria were evaluated against user's privacy preferences and if the user allowed one or another type of his profile data to be used by others to locate him, his last location along with his name, company name, title, and city/ state were displayed to the searcher (Fog. 4a). The Attendees Search Results page would tell when and where each person that matched the search criteria was seen last time. Conference Events Search service allowed to search for the events by their subject. Only conference events that were taking place at the time of the search query execution, or were scheduled to take place in the future, were returned (Fig. 4b). In addition to the event's description, location, and time, current events had an extra field containing the number of attendees of the event as tracked by IntelliBadge™ tags. Other services included mileage calculator showing individual mileage walked in the convention center along with the average, min, and max mileage walked by the rest of the attendees, and local restaurants search allowing to locate Baltimore Downtown restaurants by the type of food they serve, distance to the convention center, price, etc. Most of the functionality provided by the web kiosk application required users to login into the system and only some components (e.g., Restaurants Search) were made available to the general public.



a

b

Fig. 4. Examples of IntelliBadge™ web kiosk applications: (a) Conference Attendees Search Results page, (b) Conference Events Search Results page.

The IntelliBadge™ display application was build around 3 different visualization schemas. The purpose of the first visualization schema was to display information about current conference events (Fig. 5). The main components of this schema are:

1) Timetable showing current time, today's events and their type and location, and number of attendees at each event over time. The timetable scrolls showing only a subset of the day's schedule at a time.

2) Information Display area is used to display information about the conference. In the current implementation, it is showing the upcoming program and a map with the locations of the events.

3) Interest Profile Bars above the information display show the relative number of people from each interest category at each current activity/location.

4) Scrolling Message Bar is used for providing announcements.

5) State Table was designed as a "geek puzzle". Each badge's serial number is mapped into a unique square located in the State Table and the character inside the square corresponds to one of the tracked locations. Therefore, this table shows the location of all the badges. In addition, when a badge wearer approaches the display, the square that corresponds to his badge serial number is highlighted, thus providing him with a visual clue about the square that corresponds to his badge.

Since data about participants and conference program does not change rapidly, a continuous playback of last 3 hours was added to provide some dynamics to the display.

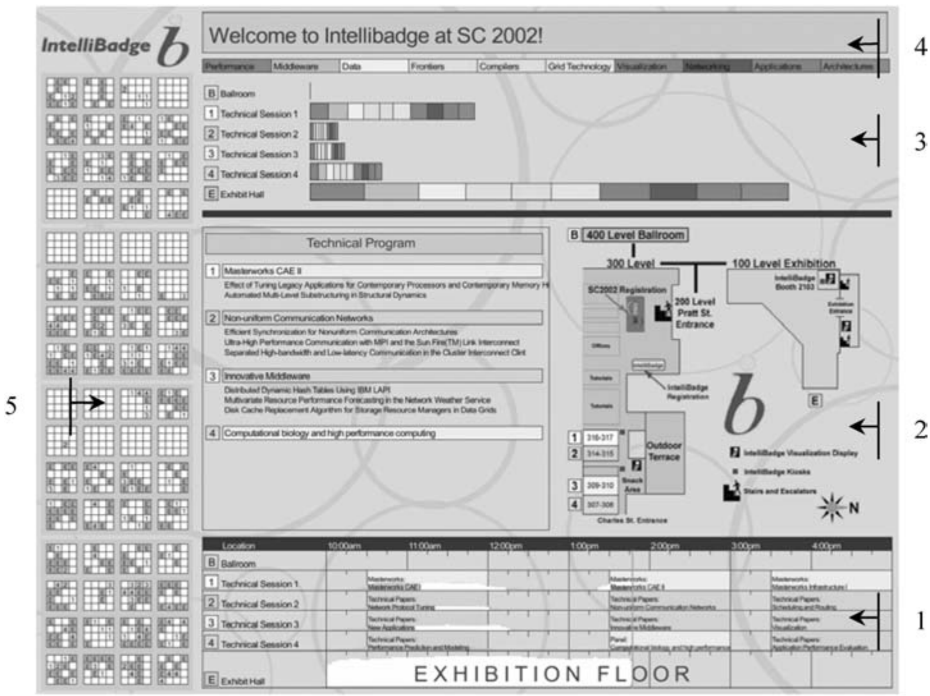


Fig. 5. "Conference-at-a-glance" visualization schema.

The purpose of the second visualization schema was to show the distribution of attendees' interest profiles in conference rooms (Fig. 6) in a poetic and artistic way. We implemented this using a garden metaphor where different tracked locations were represented as flowers in a garden. The size of the flowers corresponds to the number of people present at each tracked location at any given time. Flowers have 10 differently colored petals that correspond to 10 interest profile categories using the same coloring schema as the Interest Profile Bars in the "conference-at-a-glance" visualization. The size of each petal is proportional to the cumulative interest level for a given category based on the user profiles of the attendees present at each location. The rate at which attendees go in and out of rooms is visualized by the ants going in and out of their nest. This visualization schema was incorporated in the "conference-at-a-glance" visualization by replacing Information Display, Interest Profile Bars, and Scrolling Message Bar with the current garden image. As with the "conference-at-a-glance" visualization, a continuous playback of the past 3 hours was added to show the dynamics of the conference.

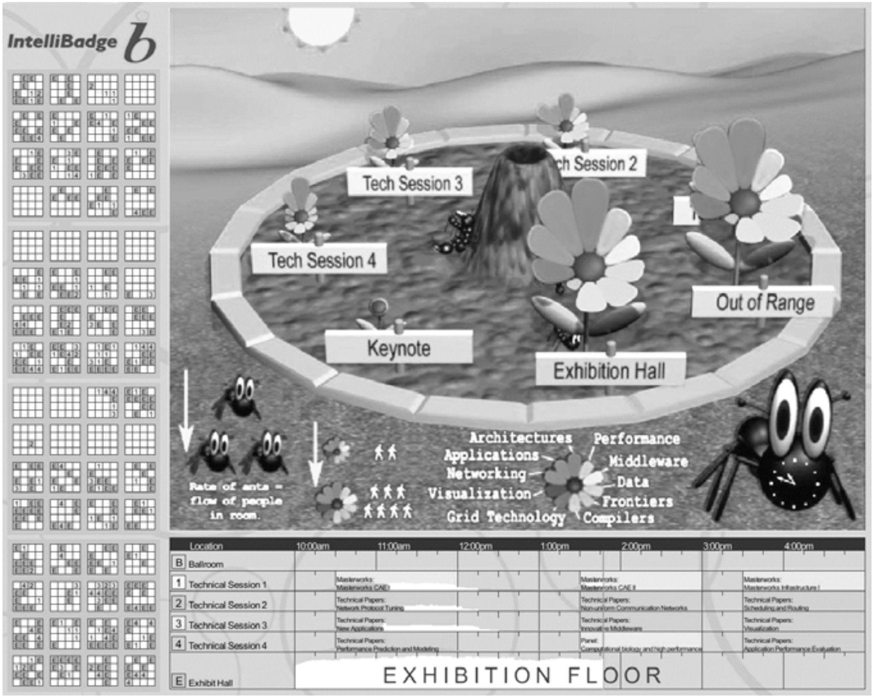


Fig. 6. "How does your conference grow" visualization schema.

The purpose of the third visualization schema was to promote social interaction among the conference attendees by revealing a non-critical personal information as they approach the display, such as their country of origin and the name of the institution (Fig. 7). A responsive international map appears and the geographical location of the participants is marked with small flags as well as the name of their institutions is shown next to it. The flag is colored-coded to indicate which visualization display the participant was standing at since all 3 displays shared the same map. In addition, the last visualization schema incorporated 3 streaming video feeds coming from the cameras installed next to the IntelliBadge™ displays. This way, participants standing at different displays could see each other. Geographical locations of all IntelliBadge™ participants were marked on the map by a small pyramid. The size of the pyramids is proportional to the number of participants coming from each unique location.

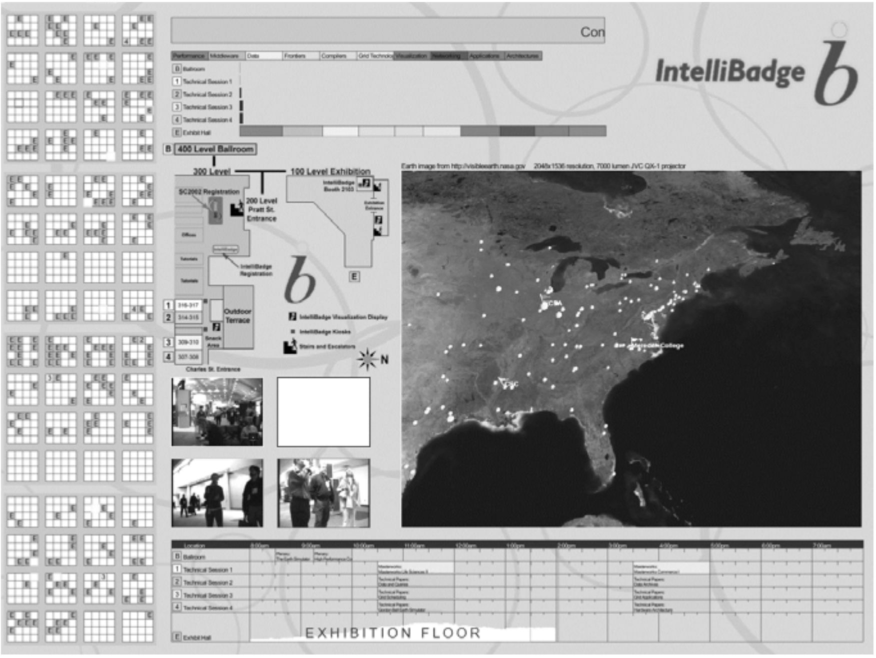


Fig. 7. Interactive visualization schema.

7 Some System Usage Statistics

In total, 890 technical program attendees registered to participate in the experiment. Eighteen participants never brought their tags to the Convention Center during the Tuesday-Thursday conference events. Thus, only 872 (40% of paid technical program registrants) were tracked at the conference. On Tuesday Nov. 19th only 857 tags were present in the Convention Center, on Wednesday Nov. 20th only 752 tags were present, and on Thursday Nov. 21st only 677 tags were present. This is because some participants returned their tags before the end of the conference, as early as Tuesday morning, and also some participants could have forgotten to carry their tags. In total, 752 tags were returned by the end of the conference: 69 tags were returned by the end of Tuesday, 69 tags were returned on Wednesday, 470 tags were returned on Thursday, and the remaining 144 tags were returned on Friday. Early tag returns are due to some attendees leaving the conference early and some attendees finding no benefits in using the provided services.

Eight hundred seventy participants provided their name during the IntelliBadge™ registration, 850 of them also allowed (revealed) others to have access to it. Table 1 summarizes what other types of information were provided and revealed. 434 participants selected Professional Title other than *Other* category and 517 participants selected Organization Type other than Other category. However, since in both cases *Other* category was the default value, it is not clear if they selected it because there was no appropriate category available or because they decided not to provide this information.

Table 1. Information provided during the registration. (1) percentage from all IntelliBadge participants, (2) percentage from IntelliBadge participants who provided this type of information.

Information type	Provided			Revealed			
	Provided	Out of	%[1]	Revealed	Out of	%[2]	%[1]
Name	870	890	97.8	850	870	97.7	95.5
Company	555	890	62.4	539	555	97.1	60.6
Address	816	890	91.7	769	816	94.2	86.4
Phone	797	890	89.6	65	797	8.2	7.3
Fax	437	890	49.1	25	437	5.7	2.8
Email	816	890	91.7	119	816	14.6	13.4
Professional title	434		48.8	842			94.6
Organization type	517		58.1	863			97.0
Interests	832	890	93.5	782	832	94.0	91.5
Group	449	890	50.5	443	449	98.7	49.8

From the time registration began (10:00am on Saturday, Nov. 16th) to the time web server was shut down (11:00am on Friday, Nov. 22nd) registered users logged into IntelliBadge™ kiosks 1771 times on-site and 1370 times remotely. Registered users logged in IntelliBadge™ kiosks on-site on average 2.2 times: 529 users used kiosks at least once, 34 users used it at least 10 times, and 8 users used kiosks at least 20 times, 361 users (41% of all registered users) newer used kiosks after the registration Fig. 8). Table 2 summarizes the usage of various kiosk services. People search, mileage, and current events search were 3 most frequently used services.

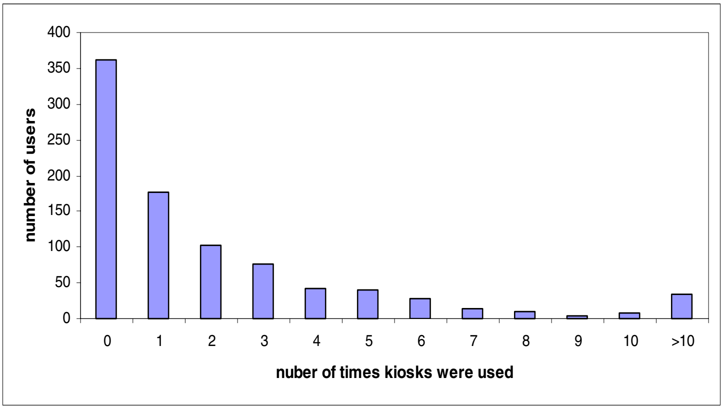


Fig. 8. Kiosks usage distribution (data is combined for the on-site and remote access).

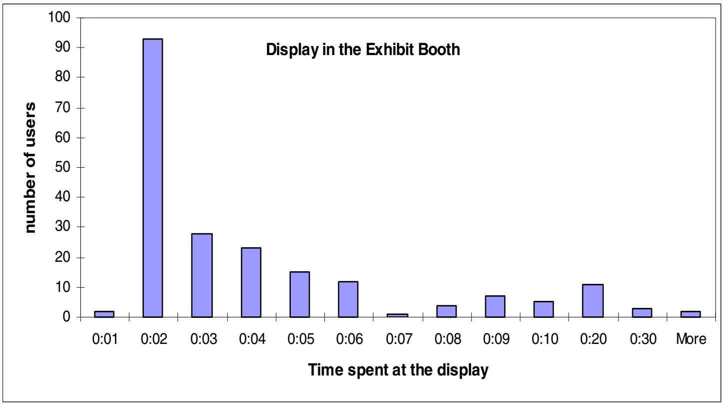


Fig. 9. Distribution of time spent at the IntelliBadge™ display in the Exhibit Hall.

On average, a user spent 3 minutes 38 seconds filling in the registration profile, an average on-site kiosk session lasted 1 minute 53 seconds, and the total time spent by all users using the kiosks (excluding the registration time) was 83 hours 7 minutes.

We consider a display to be *used* if a person spent at least 1 minute in front of it. Based on this, the display located on 3rd floor was *used* 209 times, the display at the exhibit hall entrance was *used* 37 times, and the display in the exhibit floor booth was *used* 206 times. All together 3 displays were *used* 452 times. Fig. 9 shows distribution of time spent by the participants at the display on the Exhibit floor. An average display usage session continued for 3 minutes 35 seconds combined for all 3 displays. An average display usage session for the display in the Technical Program area was 3 minutes 7 seconds, for the display at the Exhibit Hall entrance was 2 minutes 22 seconds, and for the display in the Exhibit Hall booth was 4 minutes 17 seconds.

On average, attendees spent 7 hours 55 minutes on the exhibit floor, 3 hours 58 minutes in the technical program area, and 2 hours 27 minutes attending technical program events. Out of 890 IntelliBadge™ participants, only 592 attended Technical Program sessions. On average, each conference participant attended 3.6 sessions. It should be noted that since we did not track every Technical Program registrant and did not pay any attention to the selection of the subset of the IntelliBadge™ participants, the resulting data cannot be deemed reliable to judge about the conference itself. Rather it is used here to demonstrate what kind of results can be obtained about the attendees and the conference.

Table 2. IntelliBadge™ kiosk pages usage.

Page	User profile	User Interests	Current events	Events search	People search	Mileage	Restaurants	Prizes	Info	Legal	Behind the scene	Signup
number of 'uses'	1109	548	1205	394	4261	2623	721	347	703	32	101	36

8 Lessons Learned: System Usage

Most of the SC2002 committee members and IntelliBadge™ staff could only guess how many people would participate, and they anticipated that only a couple hundred would volunteer. During the registration and with all printed material, IntelliBadge™ staff impressed upon attendees that this was an academic experiment sponsored by IEEE and that the collected registration data would be protected from distribution. We ended registration after dispensing 890 RFID tags.

The technical program ran from November 19-21 and we had planned to track people during this time. Participants did not understand this right away and wanted to find people immediately.

Participants displayed interesting behavior during the registration. They loved to create groups and get people to join those groups. Thus, an unexpected by-product of being able to create a group was the fact that people would solicit their friends to join IntelliBadge™.

The killer application turned out to be mileage calculation: people were very competitive and wanted to know how many miles they walked during the conference as compared to others. Some participants were quite unhappy if the mileage did not reflect their perceptions. Many participants returned to registration kiosks repeatedly to find out how many miles they had walked.

The Technical Program area and IntelliBadge™ Exhibit Booth received the most attendee traffic. The interactive visualization schema provided a kind of cocktail party atmosphere where people gathered, and people locating one another on the map encouraged casual conversation. Several participants met one another when their country of origin was displayed and they discovered others of like origin.

At the end of the conference attendees filled in a short conference survey that, among other questions, included the following: "If you participated in the IntelliBadge experiment, were the offered services useful and easy to use? Would you be interested in participating next year? If so, what would you add or change?" Majority of the surveyed participants provided a positive feedback and were interested in participating next year. The main suggestion by a large group of participants was to deploy more location markers so that a more precise location can be found (e.g., "It would be better if more people participated and if the sensors were more fine-grained for more detailed location"). Few complained about difficulties using the services, few complained that tags were not available or the registration queue was too long and therefore they could not register. Most of the negative feedback came with little or no explanation, e.g., "I found it totally useless" or "I didn't see the point". Some of the non-participants provided an insight into why they did not want to participate: "The point of that experiment eludes me. Who needs to be tracked?", or "No, privacy reasons. Never."

9 Lessons Learned: Technology Deployment

We conducted a study to identify the most appropriate technology that would allow us to track participants in real-time as they attend various conference events. We quickly realized that short of designing our own system, the most promising direction was to use RFID technology. Although this technology generally is not used to track people and has numerous limitations, we found that some of the commercial implementations could be tailored for our application. We conducted a formal evaluation of several such systems and decided that Savi Series 600 RFID System developed by Savi Technologies, Inc. (www.savi.com) was appropriate and economical for this project.

Initially, we intended to attach the Savi reader to the middle of the ceiling of the Ballroom. However it required power and network connection that were not immediately available. Therefore, it was decided to install the reader above one of the entry doors. It turned out however that this location was not optimal. Tags would be detected as attendees enter the room. However they would generally not be detected once the attendees disperse in the room. Therefore, the data collected from the ballroom reader was sufficient only to identify who/how many attendees attended events in the Ballroom, but not to study how much time they spent participating in the events.

The initial plan for the Exhibit Hall installation included having 3 Savi readers attached to the ceiling of the Exhibit Hall and spread evenly. On site we realized that this configuration is hard to implement since this would require routing power and network cables from remote locations. Instead, it was decided to install 4 readers above the SCinet NOC points where both the electrical power and network connectivity were relatively easily available. This also provided a better overall coverage for the exhibit hall. However, this made it difficult to identify in which part of the exhibit hall the tags were located since now tags could be picked up by more that one reader. This reader over-coverage of the conference led to an unexpectedly high number of near-duplicate entries in the database. The problem was further complicated by the cross-floor reader coverage: it is difficult to tell upon which floor a tag was located without line-by-line human interpretation of the near-duplicate database records. After a very thorough manual analysis we derived a set of rules to eliminate such semi-duplicate records. This prompted us not to use tag beaconing as a tracking mechanism, instead in our current implementation we only use tag (un)detect events.

The Savi Site Manager was the only machine to crash during the conference, and it took 10 minutes to get its replacement configured and running. During those 10 minutes, the seven Savi readers continued to receive and store beacon and signpost events from all tags at the conference. The performance characteristics of the site manager are such that the readers in the most densely packed areas of the conference (i.e., the middle of the exhibit floor) fell roughly 2.5 hours behind with events that occurred after the site manager crashed. This led to an unexpected system behavior: if a tag with a display signpost (un)detect event happened to gets its information to a reader far away in a relatively unpopulated area of the conference, then the display responded within 3-5 seconds to that tag's presence. However, if a tag happened to get its (un)detect event information to a backlogged reader first, then a user's presence at a display might not appear until long after the person carrying the tag has left. Onsite, we discussed the solution of causing the backlogged readers to clear their buffers, but since this would cause difficulties for subsequent data mining activities, we chose to let the readers catch up without losing tracking data.

We thoroughly tested various aspects of the IntelliBadge™ system using the same equipment and a similar system architecture before deploying the system. Knowing that we needed to support network sub-netting at the conference, we set up the system to span two Class B subnets. It turned out that our assumption that if Class B subnetting worked, then classless IP addressing would also work was incorrect. The four Savi readers installed in the exhibit hall, which were on a different subnet than the site manager, had a bug that prevented them from correctly utilizing their non-octet aligned netmask and routing to the site manager. Members of SCinet, the all-volunteer Scientific Computing Network group responsible for networking at SC2002, fixed this problem for us by effectively putting all seven readers and the site manager on the same subnet through a virtual LAN.

Since we did not know exactly what sort of processing power we need for the Linux servers, we assumed that a Dual Pentium III 550MHz machine with fast SCSI drives and 1GByte of RAM would be more than adequate for each server. In practice, while the machines did hold up quite well, the average system loads (as given by uptime) were consistently in the range of 4-5 with occasional spikes up to a load of 7. Our experience is that consistently high system loads in this range can often lead to runaway system load conditions in which no useful work can be performed. Even though this meltdown did not happen, the observed average system load levels suggests that for more than 900 IntelliBadge™ users and 7 readers, we either require more servers for load balancing or higher performance servers.

Acknowledgements

The success of the project was a result of the hard work of a large group of people. Valuable contributions to the system design were made by Alex Betts, Robin Dhamankar, Jon Dugan, Alex Farthing, Tony Kaap, Gaurav Kalmady, Jackie Kern, Albert Khakshour, Patty Kobel, Lorne Leonard, Stuart Levy, James Long, John Martirano, Robert Patterson, Mary Pietrowicz, Chris Pond, Jennifer Powell, Paul Rajlich, Anna Rovedo, Barry Sanders, Michael Shapiro, Alan Shih, Steve Quinn, and David Wheeler. We would also like to thank the SC2002 Organizing Committee, SCinet, and IEEE Computer Society for support of the project. In particular, we are grateful to Prof. Roscoe Giles (SC2002 Chair) and Prof. Daniel Reed (SC2002 Technical Program Chair) for their continuous support and encouragement. We are thankful to Thomas Manzagol and Jack Cowardin from Savi Technology, Inc. for Savi's contribution. Some of the equipment was also donated by JVC and Freeman Decorating Company.

References

1. Borovoy, R., Martin, F., Vemuri, S., Resnick, M., Silverman, B., and Hancock, C.: Meme tags and community mirrors: Moving from conferences to collaboration. In Proc. ACM Conference on Computer Supported Cooperative Work. (1998) 159-204. ACM New York.
2. Sumi, Y., Mase, K.: Digital Assistant for Supporting Conference Participants: An Attempt to Combine Mobile, Ubiquitous and Web Computing, in Proc. Ubicomp 2001, G.D. Abowd, B. Brumitt, S.A.N. Shafer (Eds.), Ubicomp 2001, LNCS 2201 (2001), 156-175. Springer.
3. Terry, M., Mynatt, E.D., Ryall, K., and Leigh, D.: Social net: Using patterns of physical proximity over time to infer shared interests. In Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (2002) 816-817. ACM Press.

UbiTable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces

Chia Shen, Katherine Everitt, and Kathleen Ryall

Mitsubishi Electric Research Labs
201 Broadway, Cambridge MA 02139 USA
{shen,everitt,ryall}@MERL.com

Abstract. Despite the mobility enabled by the plethora of technological tools such as laptops, PDA and cell phones, horizontal flat surfaces are still extensively used and much preferred for on-the-move face-to-face collaboration. Unfortunately, when digital documents need to be shared during collaboration, people are still mostly constrained to display surfaces that have been designed for single users, such as laptops and PDAs. Technologically there is a lack of computational support for shared digital document access, browsing, visualization and manipulation on horizontal surfaces. We believe support for such serendipitous meetings will play a critical role in future ubiquitous computing spaces. Our UbiTable project examines the design space of tabletops used as *scrap displays.* Scrap displays support kiosk-style walk-up interaction for impromptu face-to-face collaboration. Our design offers the affordances of a physical table. It provides the flexibility by allowing users to layout shared documents with desired orientation and position; at the same time it augments traditional paper-based interactions by providing a flexible *gradient* or *shades* of sharing semantics. UbiTable addresses visual accessibility vs. electronic accessibility of documents, an issue which is critical to ubiquitous environments.

1 Introduction

In this day and age of high technology, there is still an important role for face-to-face collaboration. Many meetings are on the road, and may be spontaneous events with no a priori planning. Flat, horizontal surfaces are natural for people to meet around and collaborate on. Figure 1 (a), reproduced from [4], shows two people sitting face to face at a table in the airport collaborating through their two laptops. However, the laptop screens are oriented toward their owners, and are difficult for the collaborator to view. Wouldn't it be more natural if their collaborative materials were simply laid out on the tabletop! Ad hoc, spontaneous collaborations are often seen around work places, in waiting areas at airports and train stations, as well as in cafes and lounges. People use flat, horizontal surfaces as the basis of their collaboration, with the ancillary support of tools such as laptops, PDAs, and paper documents.

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 281–288, 2003.

(a) (b)

Fig. 1. (a) Collaboration at an airport (reproduced from [4]) (b) Collaboration around an Ubi-Table

The goal of the UbiTable project is to provide efficient walk-up setup and fluid UI interaction support on horizontal surfaces. We wish to enable spontaneous, unplanned, collaboration where the participants share contents from their mobile devices such as laptops and PDAs. We draw technological underpinnings from peer-to-peer systems [4], ad hoc network protocols (such as 802.11 ad hoc mode, or Bluetooth), and existing authentication and encryption methods. Our focus is on design solutions for the key issues of (1) a model of association and interaction between the mobile device, e.g., the laptop and the tabletop, and (2) the provision of three specific *shades* of sharing semantics termed *Private, Personal and Public*. This is a departure from most multi-user systems where privacy is equated with invisibility. Figure 1 (b) shows the current UbiTable setup. Note that UbiTable also provides a variety of digital document manipulation functions, including document duplication, markup, editing, digital ink for drawing and annotation. The elaboration of these functionalities is out of the scope of this paper.

1.1 Observational Studies in Serendipitous Collaboration around the Table

Recent field observations and user studies of research prototypes have provided invaluable insights into some of the fundamental design issues and key requirements for collaboration on shared surfaces such as tabletops.

Brodie and Perry [1] carried out fieldwork observations of current mobile collaboration practices on horizontal surfaces in public spaces, including at airports, on a train and in hotels. Their key findings include: (1) The extensive use of flat surfaces and tables in mobile face-to-face collaboration, and (2) current forms of mobile face-to-face collaboration can be undemocratic and hence less effective and less productive because of the technology involved.

On the issue of orientation of contents on a table during collaborative work, Kruger and Carpendale [6] conducted an observational study on horizontal displays. One of the key findings indicates that orientation can be used to designate space according to social protocol. In groups, the group decides on the orientation. Private documents are oriented toward their owner. Orientation changes can represent a change in the privacy level for the document. For example turning a document towards someone else means that you wish for them to access or interact with it. In face-to-face meetings, participants use body language, document orientation, and gesture to transition their documents from private to public.

Greenberg et. al. [5] used their prototype called SharedNotes as a starting point for observing how people move from individual to group work, how personal artifacts and public artifacts are handled and moved back and forth. Important observations include: (1) user preference for the ability to move things back to private state after they have made them public, and (2) people would like to be able to shift their personal artifacts from private to public with intermediate shades between private and shared.

2 The Design, Usage, and Implementation of UbiTable

In this section we discuss the design for UbiTable, provide a typical usage scenario, and give implementation details. The guiding principles, supported by the observational data, in the design of UbiTable are:

1. **Shared scrap display:** (a) Simple walk-up utility for collaboration setup. (b) Easy and visible association between users, documents, and laptops. (c) Fluid content movement between laptops and the tabletop.
2. **Separation of privacy from visibility:** (a) A well defined gradient of three sharing semantics, *Private*, *Personal* and *Public*. (b) Shared interaction, with equal input capability to public documents, but owner controlled document distribution and replication.

Our goal is to provide an easily accessible scrap table that supports sharing of documents and ad hoc collaboration while maintaining user control over documents and gradations of privacy and visibility.

2.1 Shared Scrap Display Supporting Walk-Up Serendipity

UbiTable is designed for easy walk-up usage. At the same time, people collaborating around the table will temporarily *own* the table during their usage session. That is, the table can serve as a true scrap display device. Thus, we must provide means for people to feel secure while putting their content onto the table.

When a user walks up to the table and starts the UbiTable application, the laptop carries out the initial exchange of handshaking protocol with the tabletop to prevent eavesdropping from passersby. This can be done in the same fashion as the security layer in Speakeasy [4]. A short message between the table and the laptop across a trusted channel (e.g., IR) can be exchanged, and then subsequently be used to authenticate all further communication across the wireless connection among the table and the laptops. Once this exchange is completed, the laptop is connected to the table.

Figure 3 shows the UbiTable application screen on the laptop and Figure 4 is the UbiTable display on the tabletop. All interactions with the tabletop are done naturally by touching the table. When a laptop is connected to the table, an icon appears on the tabletop that represents this particular laptop. To associate the laptop with the side of the table adjacent to her, the user drags her laptop's icon into the side area of the tabletop. This side is then designated as her *Personal* space. The UbiTable mechanism used to translate physical contact with the tabletop into digital events allows us to associate the laptop and its *Personal* space with the user [3]. Thus we can uniquely identify users and give them appropriate control over documents.

2.2 Separation of Privacy and Visibility: Public, Private and Personal Areas

Most current desktop multi-user systems provide a binary notion of public and private data. Things in the shared space are equally visible and accessible by others, while private data is neither visible nor accessible to others. In most cases, private data is kept on one's own desktop or laptop, or viewed with special private viewing device [8]. Observations of people collaborating around a table with physical artifacts (such as paper documents) show well understood social protocols defining semi-private personal documents. These documents may be shared later in the meeting, used for reference purposes, or they may be personal copies of meeting records. They are located on the side of the table adjacent to their owner, and while visible to all, are oriented toward the owner to show they are personal and not meant to be accessed by others. Truly private documents are not visible at all (e.g., kept in one's briefcase), while publicly shared documents are usually placed in the center area or oriented in such a way that other participants feel comfortable accessing them.

UbiTable provides a similar gradient of *Private*, *Personal* and *Public* sharing models: *Private* data is not visible or accessible to others, *Personal* 'on-my-side-of-the-table' data is semi-private (it is visible but not electronically accessible by others), and *Public* data allows shared visibility and access. Moreover, a user maintains explicit control over the distribution and replication of his document even after the document has been placed in the public shared area. In essence, UbiTable separates the notion of privacy from visibility.

As shown in Figure 2, the display on the laptop is divided into two regions, *Private* and *Interaction Record.* The tabletop display consists of a shared public circular region in the middle, and two to N *Personal* regions at the edges. Figure 3 shows a tabletop with two *Personal* regions, one at each side. We use color or pattern as a visible means to indicate identity and ownership. The color or pattern of a user's laptop *Interaction Record* space matches that of the side of the table designated as his *Personal* space. For example, as shown in Figures 2 and 3, the laptop with striped display *Interaction Record* background is associated with the striped *Personal* side of the tabletop. Documents display an ownership bar of their owner's pattern across their top. When a user touches a document on the tabletop display, a shadow of his color or pattern appears around the document.

Only documents in the *Private* space (i.e., on the laptop) are saved on permanent storage. All documents on the tabletop remain transient, and documents in the *Interaction Record* space (the top pane on a laptop) are also transient.

Our design provides a gradation of private to public spaces. Documents can be 'published' to the *Personal* semi-private space on the table, shared in the *Public* space, or granted to other users. Users can take their documents back to their *Private* space from the *Public* space at any time. Documents retain knowledge of their owner, and transferring to a more public space (increasing the visibility and accessibility of the document) requires an explicit action on the part of the owner. In addition, by definition of *Public* space, all documents that are brought into the *Public* space from public origins, e.g., web pages from the Internet, can be accessed, copied and manipulated freely by all UbiTable collaborators.

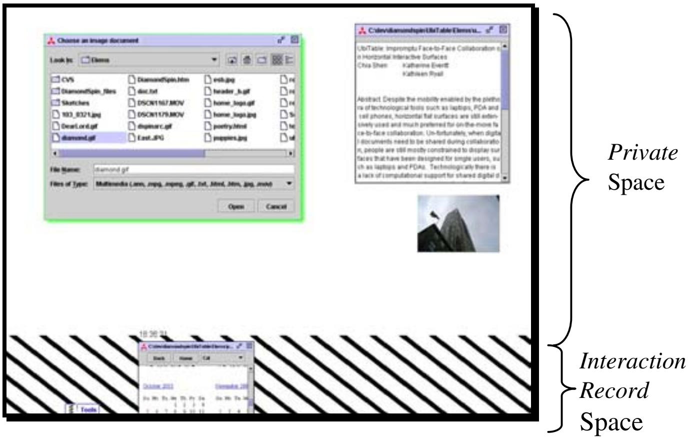


Fig. 2. Screenshot of Laptop A

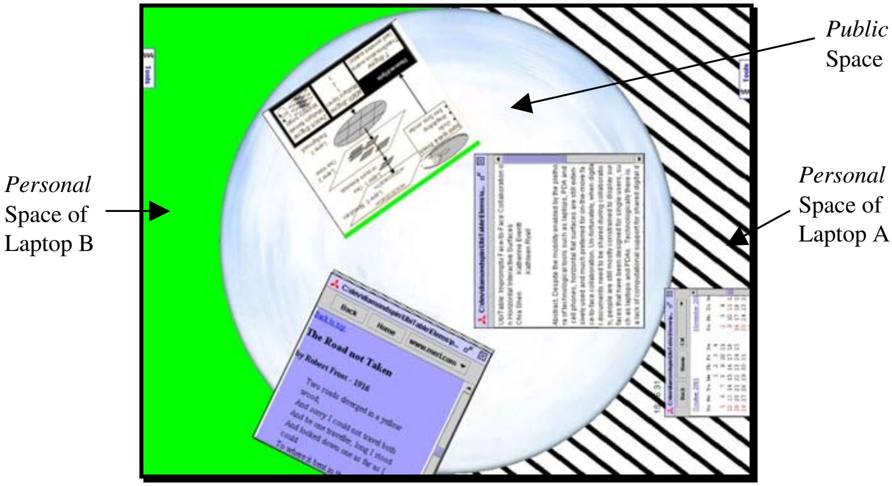


Fig. 3. Screenshot of UbiTable tabletop

The above description sounds more complicated than the actual interaction that a user will experience when using UbiTable. Both the direct manipulation style on the table and the explicit mapping of laptop color to the user color on the tabletop require very light cognitive load on the user part during interaction with the system. We believe the interaction design and visual environment allow for an intuitive process for sharing documents between the spaces, similar to natural practices of sharing physical documents.

2.3 A Usage Scenario

JoLan and Nadine meet in an airport and wish to discuss Nadine's notes for her upcoming paper. They both connect their laptops to an adjacent UbiTable. Nadine is designated pink, JoLan is designated green. Nadine opens her notes in her *Private* space, the lower pane of her laptop (colored pink). She quickly checks them for confidential data before JoLan can see them. Then she drags (using the mouse) the notes into the *Interaction Record* space, which is the top pane of her laptop screen (with a picture of the table as a visual cue). A time-stamped record of the notes document stays in her *Interaction Record* space, and the document is copied and published to her semi-private *Personal* space, the pink side on tabletop. Nadine also opens her personal calendar and publishes it to the tabletop for reference. She adjusts her personal calendar and reads off the deadline of the paper while JoLan opens a web browser within the shared space on the tabletop and locates the conference webpage. So that JoLan can interact with the notes, Nadine shares them by sliding them into the central *Public* space of the table. Sliding on the table involves pressing lightly on the electronic document and sliding one's finger to the new location. The notes document allows both users to interact with it but it maintains knowledge of its owner: it displays a pink ownership bar across the top of its frame. Nadine and JoLan take turns editing the document. Figure 4 shows the shared UbiComp webpage, JoLan editing Nadine's notes, and Nadine editing a diagram that JoLan provided. Note the pink bar at the top of Nadine's notes, indicating her ownership, and the green shadow around the notes, indicating that JoLan is currently interacting with them.

JoLan tries to take a copy by moving the notes to his green *Personal* space. The document jumps back to the *Public* space. Nadine gives him the document by pushing it into his personal corner. At the end of the collaboration, Nadine moves the document to her *Personal* side, causing a time-stamped copy to appear in her laptop's *Interaction Record* space. When they leave, the table clears all copies of their documents to preserve privacy.

2.4 Implementation

The UbiTable software is entirely written in Java. The interactive tabletop user interface is built using DiamondSpin, a freely available UI toolkit [2]. The toolkit provides facilities for continuous, arbitrary orientation of documents, managing rotation sensitive and rotation insensitive UI components, document resizing, digital ink for free-form strokes, context sensitive popup menus, as well as creating multiple virtual tabletops on the same physical table.

The underlying communication is implemented with TCP/IP over wireless LAN. The physical table (shown in Figure 2) is a DiamondTouch [3] multi-user touch sensitive surface which provides the capability to identify individual users. The DiamondTouch tabletop is connected via USB to a PC running Windows 2000. The laptops and the table communicate via ad hoc 802.11 with authentication and encryption.

3 Related Work and Summary

There have been many research projects on supporting people working together on shared displays and surfaces. The PARC CoLab project [10] enabled brainstorming meetings using prototype liveboards as shared workspaces connected with workstations. People on individual workstations can simultaneously put up their ideas on the shared workspace. The i-Land roomware project [12, 13] provides interactions on a large wall display called DynaWall, and interactive tables called ConnecTables that can be reconfigured for rapid sub-grouping in an office environment. Rekimoto in [7] explored shared continuous work space among walls, tabletop and laptops and examined the interaction techniques of hyperdragging. The system is enabled by LCD projectors and cameras which track objects tagged with visual markers, including laptops. Objects in their system can migrate freely among different computers using Java RMI and Java's object serialization mechanism. However, data that is displayed in the shared space cannot be directly manipulated and they do not deal with the ownership and visibility issues that UbiTable emphasizes. The BlueBoard project [8] is a vertical electronic touch-sensitive display[11] used as a communal large information appliance that supports both individual usage and small group sharing of personalized content. It has similar objectives as our UbiTable in terms of rapid walk-up access and simple interaction style. However, the personal content needs to be set up beforehand and stored as web-content, and accessed via a URL. Their work with transient data has inspired our treatment of data moved onto the scrap display tabletop.

The goal of our UbiTable project is to provide support for impromptu chance encounters where people need to collaborate on-the-go without prior preparation. Assuming the availability of personal laptops and PDAs, this requires approaches that can afford easy and efficient set up of shared workspaces such that users can conveniently move contents among private, personal and public spaces. The nuances between personal and public data movement resemble how people interact with paper when collaborating around-the-table. Shared social protocols offer clues to user's intention. None of the earlier research projects described above address these issues.

Our research on interactive surfaces is in some sense complementary to the peer-to-peer network, data transport, and security services that Speakeasy [4] offers. UbiTable addresses the issues of shared workspaces on horizontal surfaces, and the semantics of private, personal and public data access and exchange. Speakeasy can be used as our underlying ad hoc peer-to-peer discovery protocol and network service infrastructure.

In summary, we have presented our design and implementation of UbiTable, an interactive horizontal surface where people can walk up, share their content, and exchange data. This ubiquitous computing environment supports impromptu face-to-face collaborations for small groups of people, and provides the best of both physical and digital worlds. By designing our system to exploit many of the social protocols already commonly used in tabletop environments, UbiTable provides a comfortable and familiar interface.

References

1. Brodie, J., Perry, M.,: Mobile Collaboration at the Tabletop in Public Spaces, ACM CSCW 2002 Workshop on Co-located Tabletop Collaboration: Technologies and Directions. New Orleans, Louisiana, USA, November 2002.
2. DiamondSpin. http://www.merl.com/projects/diamondspin.
3. Dietz P., Leigh D., DiamondTouch: A Multi-User Touch Technology. Proceedings of UIST'01. Orlando, Florida. Nov 11-14, 2001. pp. 219-226
4. Edwards, W.K., Newman, M.W., Sedivy, J.Z., Smith, T.F., Balfanz, D., Smetters, D.K., Wong, H.C., Izadi, S.,: Using Speakeasy for Ad Hoc Peer-to-Peer Collaboration. Proceedings of ACM CSCW 2002, November 16-20, 2002, New Orleans, Louisiana, USA.
5. Greenberg, S., Boyle, M., Laberge, J.,: PDAs and Shared Public Displays: Making Personal Information Public, and Public Information Personal. Personal Technologies, Vol. 3, No.1, March 1999. Elsevier.
6. Kruger, R., Carpendale, S.,:Orientation and Gesture on Horizontal Displays. UbiComp 2002 Workshop on Collaboration with Interactive Walls and Tables. Gotenborg, Sweden.
7. Rekimoto, J., Saitoh, M. Augmented surfaces: a spatially continuous work space for hybrid computing environments. Proceedings of CHI '99. Pittsburgh, PA, May 1999, ACM Press, pp. 378-385.
8. Russell, D.M., Gossweiler, R.,: On the Design of Personal & Communal Large Information Scale Appliances. Proceedings of UbiComp 2001. Atlanta, GA U.S.A.
9. Shoemaker, G.B.B., Inkpen, K.M., Single Display Privacyware: Augmenting Public Displays with Private Information. Proceedings of ACM CHI 2001.Seattle WA.
10. Stefik, M.: The PARC CoLab Project. http://www2.parc.com/istl/members/stefik/colab.htm
11. SmartBoard. http://ww.SMARTTech.com/
12. Streitz, N., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, W., Reischl, W., Rexroth, P., Seitz, P., Steinmetz, R., i-LAND: An interactive Landscape for Creativity and Innovation. Proceedings of CHI '99. Pittsburgh, PA, May 1999, ACM Press, pp. 120-127.
13. Tandler, P., Prante., T., Muller-Tomfelde, C., Streitz, N., Steinmetz, R., "ConnecTables: Dynamic Coupling of Displays for the Flexible Creation of Shared Workspaces", Proceedings of ACM UIST'01. Orlando, Florida, November, 2001.

Social Network Computing

Nathan Eagle and Alex (Sandy) Pentland

MIT Media Lab, 20 Ames St, Cambridge, MA 02319
{nathan,sandy}@media.mit.edu
http://reality.media.mit.edu

Abstract. A ubiquitous wearable computing infrastructure is now firmly entrenched within organizations across the globe, yet much of its potential remains untapped. This paper describes how the handheld computers and mobile phones in today's organizations can be used to quantify face-to-face interactions and to infer aspects about a user's situation, enabling more creative and transparent functioning of human organizations.

1 Introduction

Ubiquitous wearable computing has arrived in today's knowledge organizations. Handheld computers and mobile phones have been adapted as standard corporate attire across the globe. And the potential functionality of this new business uniform is dramatically increasing. Personal Digital Assistants (PDAs)[1], once computationally limited to storing calendar and contact information, now have wireless network connectivity and run at the speeds comparable to the desktop computers just a couple years ago. There are thousands of organizations comprised of millions of individuals who currently carry wireless transceivers, microphones, and three times the computational horsepower of an Intel Pentium I processor *in their pocket*.

Parallel to this wearable computing infrastructure lies what this paper refers to as an organization's social infrastructure. Although the traditional 'org chart' is meant to reflect the scaffolding of a social infrastructure, hierarchical job titles do very little to characterize an organization's underlying complex human network. More indicative of social infrastructure is the wisdom accumulated throughout an employee's extended career within an organization, for example: learning which people really influence results, who are the true experts on a subject, which people work well together, or who should connect with whom.

The ubiquitous computing infrastructure within the workplace has the potential to augment organizational functioning by making social infrastructure more transparent. Using custom analytic software and a mobile computing infrastructure consisting of thirty wireless, linux-based handheld computers and an 802.11b network, we have created a testbed that we are using to learn how to make organizations that are more creative, efficient, and open.

---

[1] Throughout this paper, PDA is used interchangeably with the term 'handheld computer'.

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 289–296, 2003.

2 Quantifying Face-to-Face Interactions within the Workplace

The social network research community has used survey data almost exclusively to establish the relationships between individuals within organizations. Despite the introduction of web-based surveys or experience sampling methods using surveys on handheld computers [7], the fundamental flaws inherent in self-report survey data remain: data bias and sparsity.

With the advent of corporate email and instant messaging, behavior measurement techniques are augmenting the surveys, enabling new social network datasets that require less direct participation from the participants. Similarly, telephone logs can be analyzed to gain insight into the relationships between coworkers.

Quantifying the face-to-face interactions within an office environment is of particular interest, especially because complex information is rarely transmitted in an office environment by any other means [1]. If an individual requires a complex piece of knowledge from a colleague, he would use the telephone or email to set up a meeting, but then receive the information through a face-to-face interaction. Even outside the context of meetings, informal face-to-face conversations in the hall or by the water cooler are incredibly important for organizations [5]. Effectively harnessing this face-to-face communication channel has the potential to revolutionize the field of knowledge management.

Previous work at quantifying face-to-face interactions has been mainly with 'badges' that use infrared and RF to track individuals and their meetings. Choudhury and Pentland, for instance, built a shoulder-mounted 'sociometer' that incorporated IR, sound level and accelerometers in order to track interactions within an organization [4].

To move from social network mapping to social network function, we must also capture information about discussion content and context. Our system accomplishes this by capturing and analyzing each participant's audio, annotating the audio with subjective user feedback, extracting keyword-based topic and context information, and using audio and 802.11b data to establish location and other participants in local proximity. As shown in Table 1, an analysis of synchronized audio streams from coworkers can provide insight into individual social behavior as well as the efficiency of the collective.

3 The Reality Mining System

Mirroring the ubiquitous wearable computing infrastructure in the modern workplace, the Reality Mining system is a combination of commercial hardware running specialized software. Thirty 802.11b-enabled PDAs containing standard personal information management applications were augmented with the ability to continuously stream and store audio and establish the proximity of others. The largest benefits of the system are realized as it scales. Detailed information regarding the dynamics of the face-to-face communication within the workplace can be quantified and correlated with the roles individuals play in an organization's social infrastructure [2].

The heart of the Reality Mining system is the Sharp Zaurus. These linux-based, 206 MHz handheld computers were equipped with 802.11b CF cards and 256 MB of storage. Audio was captured from a variety of wired and wireless mobile phone headsets and lapel microphones that connect to the Zaurus through its audio jack. For all day use, interchangeable 1850 mAh battery packs were plugged into the AC adapter jack.

Applications for the Zaurus were created to record audio continuously, storing it locally until it could be transmitted to a server over an available 802.11b network. Besides streaming audio, packets in this wireless network could be 'sniffed' by the PDAs interested in determining who else is in the local proximity. Information regarding access point signal strength information was correlated with location using a static table look-up procedure. More interactive applications were written for meeting analysis that accumulated and displayed aggregate interest statistics in real-time. On the server-side, conversation detection, analysis, and inference software were written to process multiple large audio files in parallel[2].

Conversation Detection. Our speech detection algorithm incorporated a variation of a multi-band center clipper. Each audio stream is chunked and run through a bank of filters in the frequency domain. The output energy is thresholded to generate tentative speech segment labels (talking / not talking) over each second. An error-checking script correlates waveform segments over a short window to verify that the labeled speech regions were not due to another participant speaking loudly.

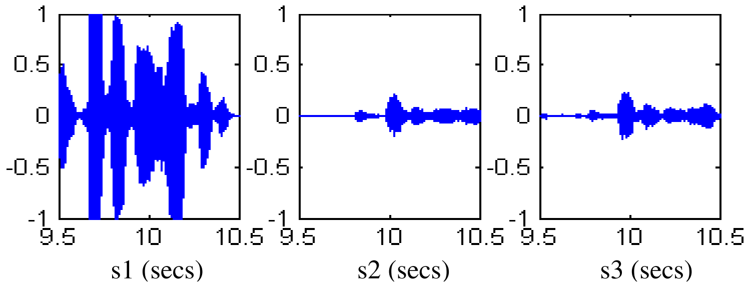


Fig. 1. These three waveforms sections are strongly correlated (*c*>*.25*), indicating that s2 and s3 are within earshot of the speaker (s1). An analysis of the relative energy between the waveforms can yield insight into the physical proximity of the speakers.

Establishing accurate vocalization labels is only the first part of conversation detection. The next step is to determine the proximity of the participants. This can be initially accomplished by comparing the access points to which the participants are streaming audio. The audio segments of participants who are near similar access points are then correlated to determine whether they are within earshot of each other.

[2] Streamed at 22KHz 16-bit, each person has a daily audio file of approximately 1 GB.

Essentially the concept uses the principle that a speaker's voice will not only be recorded in his own microphone, but also at a lower energy level in the microphones of the people around him. However, if there is a high correlation between two audio segments, this is not yet substantial evidence of a conversation. A correlation would occur even when two adjacent users are having separate conversations on mobile phones, or separate dialogues with individuals not using the Reality Mining system. As shown in Figure 2 and described in [3], the voicing segment of one participant is the noisy compliment of the other participants' voicing segments. Measuring the mutual information between these binary streams has been shown to be indicative of an actual conversation[3].

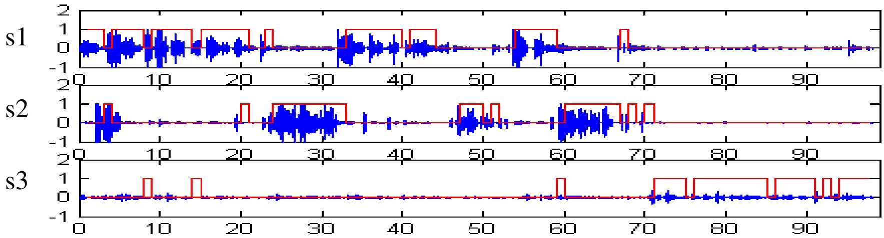


Fig. 2. The three speaking segments have high mutual information (*MI>.3*), indicative of a conversation between the speakers. It can be see that each voicing segment is the noisy complements of the remaining two.

Conversation Analysis. Once detected, the audio streams of a conversation are extracted and analyzed. Table 1 shows a selection of features that can be gleaned from this audio data. Profiles of a participant's typical social behavior are built over time using conversation features such as speaking rate, energy, duration, participants, interruptions, transition probabilities, time spent holding the floor [3], and annotated by interest metrics. By comparing relative volume levels of a speaker's voice in multiple microphones, it even becomes possible to infer proximity of the participants to an approximate degree.

Throughout the meeting, the PDAs were serving a dual purpose. While streaming audio and wireless network information to a central server, the handheld computers were also enabling a user to input his or her interest level through an interface designed to minimize distraction.

This type of analysis allows objective assessment of an individual's influence and contributions to the meeting, as well insight into the effectiveness of the group. Feedback to the most vocal speakers can be used to encourage them to share the floor with others. The more soft-spoken participants whose comments are appreciated by the group now have a means of receiving recognition of their contribution. Patterns in the behavior of dyadic pairs over time can be even more telling. Information about the

---

[3] Mutual information was also initially used to calculate approximate alignment between the audio streams.

people who interrupt, sit next to, or yield the floor to others, provides data that can be directly correlated with relationship. Using the topic-spotting methods described below, an individual's influence can also be correlated with how the group incorporates the topics popular with the individual.

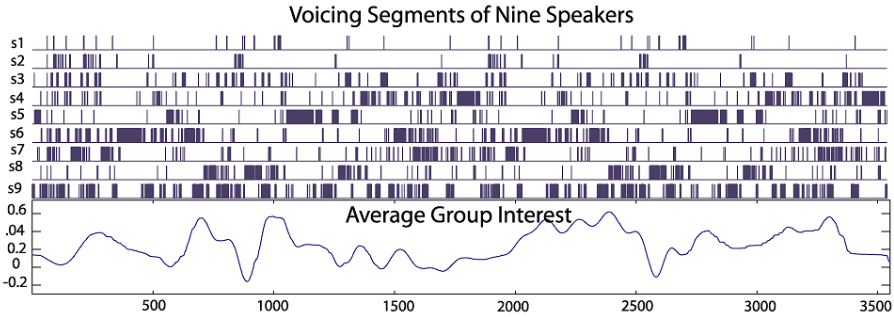


Fig. 3. The voicing segments and interest level of a one-hour meeting.

Table 1. Meeting Analysis. A one-hour meeting in which the participants were wearing the Reality Mining system, streaming audio to a central server and simultaneously recording their interest level.

Participant	Speaking time (%)	Avg (sec) Comment	Nearest Neighbor	Transition (Name, %)	Avg Interest	Group Interest
s1	1.5	4.1	Nathan	Nathan-27	.21	.44
s2	2.2	2.2	Sandy	Sandy-47	.13	.36
s3	9.9	3.5	Sandy	Jordan-22	.20	.22
s4	11.4	9.6	Mike S	Mike O-23	.05	.30
s5	12.8	8.8	Mike S	Sandy-37	.18	.33
s6	16.9	6.6	Jordan	Mike S-28	.09	.21
s7	10.1	6.6	Jordan	Sandy-30	.19	.24
s8	10.8	10.9	Ivan	Sandy-26	.40	.32
s9	24.4	6.9	Mike S	Mike O-22	.17	.25

Content and Context. The final component of the audio analysis is establishing the conversational situation: the topic and the surrounding context of a conversation. ViaVoice, a commercial speech recognition engine, is used to transcribe the audio streams, however its transcription accuracy often falls well below 40% for spontaneous speech recognition. For situation understanding, our system combines a network of commonsense knowledge with keywords and contextual information automatically obtained from the Zaurus. We make use of Push Singh's OpenMind network, containing over 250,000 commonsensical semantic relationships contributed from over 10,000 people across the web [8]. Despite this vast amount of data, the knowledge database can be compressed into fewer than 50 MB and easy stored locally on the PDAs. While the words the speech recognition engine gets correct tend to be grouped around neighboring semantically-related nodes, errors in the transcriptions turn out to be distributed randomly over this network. The nodes surrounding the largest clusters of keywords are assumed to be potential aspects of the speakers' situation. However, the robustness of the classifier comes from its ability to bias the prior probability of each node based on other contextual information from the PDAs, such as the user's location, conversation participants, or simply the people in his local proximity. Online learning algorithms incorporate subsequent observations into the classifier yielding a specialized model that better reflects an individual's behavior.

With only one correct word for every three, even a human would have a difficult time inferring the gist of a transcribed conversation. But just as additional contextual and common sense information can help a human infer the topic of a conversation, this type of information can be equally beneficial to a probabilistic model. Given a commonsense knowledgebase, along with contextual information from these mobile devices, creating a classifier to determine gist of noisy transcriptions becomes tractable [6].

Table 2. Conversational Inference. Two participants were standing in line, talking about what to order in the food court cafeteria. The situation classification with only the noisy transcript is shown in Table 2.1. Table 2.2 incorporates additional contextual information: the fact that the audio was streamed to the food court access point.

Table 2.1

Confidence	Classification with no context
5	Eat in restaurant
5	buy beer
5	talk with someone far away
5	eat in fast food restaurant
5	buy hamburger
4	go to hairdresser
4	wait in line

Table 2.2

Confidence	Classification with location context
27	eat in fast food restaurant
21	eat in restaurant
18	wait on table
16	you would go to restaurant because you
16	wait table
16	go to restaurant
15	know how much you owe restaurant

4 Applications

Once pocket-sized devices become more aware of the infrastructure in which they are part, a variety of exciting applications become possible. Three applications that are now being evaluated in classes at MIT include:

Meeting Miner: Participants continuously provide subjective feedback on comments and discussion using a 2D touch pad. The feedback interface converts the task of providing continuous feedback into a low-attention, secondary task. By correlating peaks in interest/approval with the individual audio inputs, the system can automatically provide a summary audio track consisting of comments that had high approval or interest ratings, and to employ speech analysis to identify topics that had high (or low) ratings.

OpinionMetrics: Subjective feedback is pooled and shared with the participants via a public display. Comments that give rise to wide variations in opinion cause the discussion to focus on the reason for disparate opinions, and controversial topics can be retrieved for further analysis and debate. Opinions and comments can also clustered using ‘collaborative filtering’, to display groupings of opinion, allowing within-group and between-group debate

GroupMapper: Dynamic maps of social infrastructure can be generated and publicly displayed to reflect the roles and dyadic relationships that individuals have within a work group. It is hoped that such analysis will help with such tasks as determining who to ask for help, identifying isolated cliques, and a gaining a deeper insight into the underlying dynamics of the organization. Architects have expressed interest in using this system to monitor how small changes to the interiors of buildings have an effect on the office communications.

Privacy: Although our system uses encryption and permissions to address some problems of privacy, significant concerns remain. To deploy the system at the enterprise level, additional work needs to be spent to assuage the concerns of the more privacy conscious. Quick modifications will include a ten-minute delete and temporary mute button. Another potential modification to the system would be to stream the audio to the participants’ personal computers. At the end of each week, the conversation inference algorithms could be used to summarize a user’s interactions, creating a list of the week’s conversations including location, topic, people in proximity, and duration. Along with each interaction, would be a checkbox to mark if the conversation is private, public or should be permanently deleted. The ability to have weekly interactions quantified and displayed can provide insight into an individual's personal time management and may create enough value to justify the system on its own. This could be especially important for organizations of individuals who need to keep careful track of how they spend their time for billing purposes.

5 Future Research and Conclusion

This project demonstrates our ability to capture extremely rich data on everyday human behavior, including interactions with others, movement, location, and activities, using hardware already worn daily by millions. We are now instrumenting group activities such as negotiation, brainstorming, and weekly group meetings to derive relationship information, and using this information in controlled experiments to measure the extent to which it can be leveraged to create more effective teams and collaborations.

Such a data-driven model of social network functioning offers the potential to transcend the traditional org-chart, perhaps by drawing parallels to ad-hoc network optimization. Forming groups based on heretofore unrecognized inherent communication patterns rather than an orthodox hierarchy may yield significant insights to the organizational structure community. We believe that modern organizations will inevitably try to leverage their existing ubiquitous wearable computing infrastructures. Our system provides a testbed and baseline to build and demonstrate future social network applications.

Acknowledgements

Thanks to Jonathan Gips for his work developing the Zaurus interest poll application and to Push Singh for his help working with the OpenMind Network. This work was partially supported by the NSF Center for Bits and Atoms (NSF CCR-0122419).

References

1. Allen, T. J.: Organizational Structure for Product Development, Working Papers. (2002)
2. Allen, T. J.: Managing the Flow of Technology, MIT Press, Cambridge, MA (1977)
3. Basu, S: Conversation Scene Analysis, in Dept. of EECS. Cambridge: MIT (2002)
4. Choudhury, T., Clarkson, B., Basu, S., and Pentland, A.: Learning Communities: Connectivity and Dynamics of Interacting Agents. Proceedings of the International Joint Conference on Neural Networks. Portland, Oregon. (2003)
5. Cohen, D., Prusak, L.: In Good Company: How Social Capital Makes Organizations Work, Harvard Business School Press, Cambridge, MA (2001)
6. Eagle, N., Singh, P., Pentland, S.: to be apear in the Artificial Intelligence, Information Access, and Mobile Computing Workshop at the 18th International Joint Conference on Artificial Intelligence (IJCAI). Acapulco, Mexico. (2003)
7. Feldman Barrett, L., & Barrett, D. J.: Computerized experience-sampling. Social Science Computer Review, 19, 175-185. (2001)
8. Singh, P.: The public acquisition of commonsense knowledge. Proceedings of AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access. Palo Alto, CA: AAAI. (2002)

The Design of a Context-Aware Home Media Space for Balancing Privacy and Awareness

Carman Neustaedter and Saul Greenberg

University of Calgary
Department of Computer Science
Calgary, AB, T2N 1N4 Canada
{carman,saul}@cpsc.ucalgary.ca

Abstract. Traditional techniques for balancing privacy and awareness in video media spaces, like *blur filtration,* have been proven to be ineffective for compromising home situations involving a media space. As such, this paper presents the rationale and prototype design of a context-aware *home media space (HMS)*—defined as an always-on video media space used within a home setting—that focuses on identifying plausible solutions for balancing privacy and awareness in compromising home situations. In the HMS design, users are provided with *implicit* and *explicit control* over their privacy, along with *visual* and *audio feedback* of the amount of privacy currently being maintained.

1 Introduction

A *home media space* (HMS) is an always-on video-based media space used within a home setting. It is designed specifically for the telecommuter who chooses to work at home, but who still wishes to maintain a close-working relationship with particular colleagues in remote office environments. Like all media spaces, the video provides the telecommuter with awareness information about their collaborator's availability for conversation, and a way to easily move into casual communication over the same channel. Unlike office-based media spaces, a home media space has to pay considerably more attention to how the system appropriately balances privacy and awareness, because privacy concerns are far more problematic for home users.

In this paper, we describe the rationale and prototype design of our own context-aware home media space. Specifically, we detail how and why:

1. existing privacy mechanisms are leveraged for use in home-based video conferencing systems;
2. implicit actions using context-aware technology can regulate privacy;
3. no implicit action will ever decrease the amount of privacy without first warning the user and providing the opportunity to stop the operation;
4. explicit actions using dedicated physical controls and gesture recognition can regulate privacy; and,
5. visual and audio feedback makes the state of the system easily discernable at any time.

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 297–314, 2003.

We begin by briefly describing our motivation: casual interaction and informal awareness. Next, we outline the privacy concerns that can arise from telecommuting, and how previous work suggests that context-aware computing offers solutions for balancing privacy and awareness in a HMS. Finally, we discuss the rationale and design of our context-aware HMS and the privacy-protecting strategies it offers.

2 Casual Interaction *vs.* Privacy in Home Telecommuting

To set the scene, this section briefly summarizes the importance of casual interaction and awareness. We describe how video-based media spaces can provide rich awareness for distance-separated telecommuters, but at the expense of privacy violations.

2.1 Casual Interaction, Awareness, and Media Spaces

Throughout a typical day, co-workers naturally interact amongst each other in what is known as c*asual interaction:* the frequent and informal encounters that either occur serendipitously or are initiated by one person [11, 16]. These interactions have been shown to foster knowledge and help individuals accomplish both individual and group work [11, 18]. Casual interaction is held together by *informal awareness*: an understanding of who is around and available for interaction. It is this awareness that helps people decide if and when to smoothly move into and out of conversation and collaboration [18]. Informal awareness is easily gained when people are in close physical proximity, but deteriorates over distance [13, 18]. As a result, casual interaction suffers when co-workers are distributed.

One possible solution for providing awareness between distance-separated collaborators is the *media space*: an always-on video link that connects remote locations [5, 10, 11, 13, 14, 19, 20, 23]. Its advantage is that the always-on video channel can provide rich awareness in a manner that is easily understood by individuals. In practice, video media spaces have found some limited success in office situations, albeit primarily at research laboratories (e.g., 11, 17, 20). The problem is that these media spaces also broadcast information that individuals may consider to be privacy sensitive [4, 5, 6, 14, 16].

In an effort to help mitigate privacy concerns over video links, researchers have studied many techniques, with one of the most popular being *distortion filters:* algorithmic reduction of image fidelity that hides sensitive details in a video image while still providing awareness [6, 14, 21, 25]. Other researchers have tried similar techniques where alternate images are presented in place of actual video frames [8, 16]. Distortion filters have proven successful at balancing privacy and awareness for mundane and benign office situations, e.g., people working or reading, people chatting, people eating lunch [6].

In spite of this (and other) research, most media space installations simply ignore privacy issues. There may be several reasons for this: risks are fairly low in office settings; installations are between close colleagues or early adopters; and simple privacy safeguards often suffice, e.g., people can explicitly switch off the video channel, or turn the camera around to face the wall.

2.2 Privacy in Home-Based Media Spaces

With the declining cost of PC cameras and several companies offering free video conferencing software (e.g., Webcam for MSN Messenger, Yahoo! Messenger), video is increasingly being used in the home. Its prevalence is indicated by the growing number of live webcam sites on the Internet.

Privacy concerns become complicated when people choose to work from home as telecommuters, while still desiring close contact with colleagues at work. The big problem is that privacy risks increase drastically for the telecommuter (compared to the office worker), as well as for others in the home. Privacy threats increase for several reasons:

- ***The home is inherently private in nature.*** Normally people are more relaxed at home and able to deviate from social customs [1]. This makes it easier for people to do unconscious acts such as picking one's nose, scratching one's rear, or other potentially embarrassing actions that can be inadvertently captured by the camera.

- ***The telecommuter lives a dual role as worker and home occupant.*** Appearances and behaviours that are appropriate for the home may not be appropriate when viewed at the office. For example, it is appropriate for a telecommuter to work at home shirtless or in pajamas, yet the same level of dress may not be appropriate when seen at the office and may violate the telecommuter's privacy.

- ***The dual purposes typical of most home offices.*** The home office may also be a corner of a living room, or a spare bedroom. Unknowingly, home occupants may be caught on camera in precarious situations as a result. For example, a house guest may be using the home office/spare bedroom in the evening when the camera accidentally captures her changing clothes (because the "owner" may have forgotten to either warn the guest or turn off the camera).

- ***Threat/benefit disparity.*** Individuals in the home who may gain little or no benefit from the HMS still incur its privacy threat. For example, a spouse who does not want to be captured on camera may be recorded just by simply entering the home office. This situation could be quite privacy-sensitive if (say) the spouse came in to the home office to kiss his or her mate!

These increased privacy risks suggest that home media space systems must incorporate techniques that somehow mitigate privacy concerns. Of course, one possibility is to simply adapt techniques already proposed for office media spaces. Unfortunately, most have not been tested for "high risk" situations such as those arising in the home. Consequently, as motivating work for our current research, we evaluated *blur filtration*—a distortion technique that produces a blurred video image—for its effectiveness in balancing privacy and awareness for compromising home situations [21].

In our study, people were shown video scenes ranging from little risk to extreme risk. Each scene was first shown extremely blurred, with subsequent showings less blurred until eventually the scene was shown in full fidelity. We looked for the thresholds at which people could just extract awareness information from the scene, and also the thresholds at which people would judge as violating privacy. The results clearly showed that blur filtration is not able to balance privacy and awareness for risky home situations, i.e., the level of blur that let people garner just enough information to judge someone's availability was above what people felt was 'safe' to show others. Our study also showed that as privacy risk increases, people begin to abandon filtration as a strategy for preserving privacy and choose to simply turn off the camera. People simply do not trust techniques where the camera remains facing them during risky situations, as is the case with many of the strategies that have been studied to preserve privacy for office-based media spaces. Rather, people prefer direct control of their privacy, e.g., being able to position the camera, control the blur level, turn the camera on/off, and so on.

3 The Design Philosophy of Our Context-Aware HMS

This section outlines the five principles behind the design of our context-aware HMS. First, we provide background knowledge of social psychological theories of privacy mechanisms. Second, we use this knowledge to explain each of our design principles and why they are included in our design philosophy. Third, to set the scene of our design, we describe the design elements that arose from our five principles.

3.1 Design Principles for a Context-Aware HMS

The results of our study on blur filtration [21] highlighted the importance of providing user control over information conveyed through a video media space. To provide natural mechanisms for users to control this information, we began investigating how humans regulate privacy in everyday life through various behaviours and actions called *privacy mechanisms* [2]. We will use the terms "privacy mechanisms" and "privacy-protecting strategies" interchangeably for the remainder of this paper. Each and every culture has used privacy mechanisms to regulate interaction with others [2]. When individuals require more privacy, they use these mechanisms to let others know they desire less interaction. Just the same, when individuals require more interaction, they use these mechanisms to let others know they desire less privacy. These privacy mechanisms are very natural and often form an unconscious act [1]. The privacy mechanisms used by humans can be classified into four categories [1]:

1. ***Verbal behaviours***: the use of the content and structure of what is being said;
2. ***Non-verbal behaviours***: the use of body language, e.g., gestures and posture;
3. ***Environmental mechanisms***: the use of physical artifacts and features of an environment, e.g., walls, doors, spatial proximity, timing; and,
4. ***Cultural mechanisms***: the use of cultural practices and social customs.

Research has shown that different cultures employ mechanisms from different categories [2]. Western culture typically relies on environmental mechanisms (e.g., the physical architecture of our homes), whereas other communal cultures rely more on cultural mechanisms (e.g., when and where people gather in a house).

Based on this research, we believe the design of a HMS should use the following design principles:

1. existing privacy mechanisms should be leveraged for home-based video conferencing systems;
2. implicit actions using context-aware technology can regulate privacy;
3. no implicit action should ever decrease the amount of privacy without first warning the user and providing the opportunity to stop the operation;
4. explicit actions using dedicated physical controls and gesture recognition can regulate privacy; and,
5. visual and audio feedback makes the state of the system easily discernable at any time.

The first principle helps to create privacy mechanisms for a HMS that are both easy to understand and natural to use because they are based on techniques already familiar to humans. Our design supports this principle by providing users with privacy-protecting strategies from the same four categories used by humans in everyday life (discussed in more detail later).

Privacy regulation in real life is lightweight and often transparent. Such implications should also be available to HMS users. Thus, as the second principle states, privacy-protecting strategies in a HMS should also be lightweight and transparent. Our design supports this principle by using context-aware computing as a tool for balancing privacy and awareness through implicit means. Unlike previous work in context-aware computing [22, 24], we enable one specific location—a home office/spare bedroom—with technology that senses who is around and then infers privacy expectations through a simple set of rules.

There is still a considerable gap between human expectations and the abilities of context-aware systems [9]. Context-aware systems can make mistakes and it is important that these mistakes do not increase privacy threat; the third design principle addresses this problem. Our design supports this principle by first warning users that an implicit action has initiated a privacy decreasing operation; and second, by providing an opportunity for users to override this operation.

The fourth principle also addresses the previously mentioned problem by recognizing that we need to keep the user in the "control loop." Our design supports this principle by providing users with dedicated physical and graphical controls, where explicit actions such as adjusting a physical slider or gesturing towards the camera will alter the privacy level. We recognize that explicit control must absolutely be lightweight and executed with almost trivial effort.

The fifth principle is important because users must be able to fine tune the privacy/awareness balance as desired. To do this fine tuning, they must know how much privacy is currently being maintained. Our design supports this principle by providing feedback of the achieved privacy level through audio and visual cues, rendered on both physical displays (such as LEDs) and on the screen. This feedback is both understandable and continually available.

3.2 Elements of a Context-Aware HMS

To foreshadow the details of our design, this section outlines the elements of our HMS that arise from the five design principles. The subsequent section describes their importance by outlining how they work together to regulate privacy.

Figure 1 shows the HMS's graphical user interface (GUI) as seen by the telecommuter: the top window shows a mirrored image of the telecommuter as it is captured, and the bottom window shows the telecommuter's colleague. A third window contains additional options (Figure 2) and is displayed by clicking the configuration button in the telecommuter's toolbar (Figure 1, top, fourth button from the left). The other graphical controls are described below. Figures 3 and 4 show the layout of the HMS in the home office/spare bedroom of a telecommuter. The design is specific to this room layout, but the ideas presented can be applied to a variety of home settings.

We support the five HMS design principles, discussed previously, by including specific elements within our design:

Fig. 1. The HMS GUI: the telecommuter (top) and colleague (bottom).

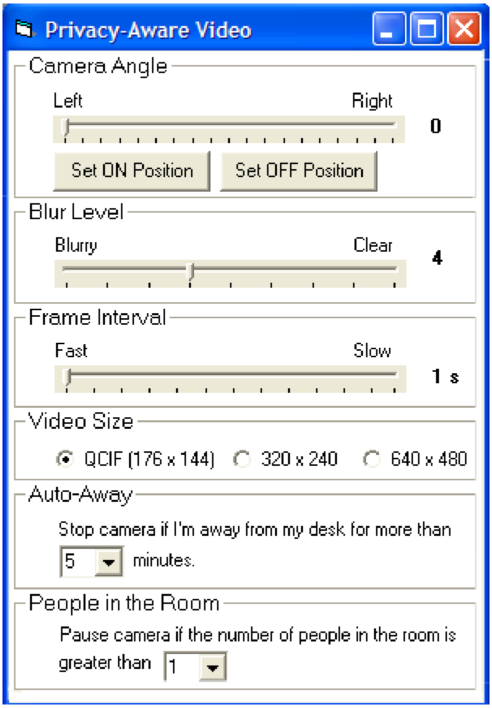


Fig. 2. A configuration window to adjust various HMS attributes.

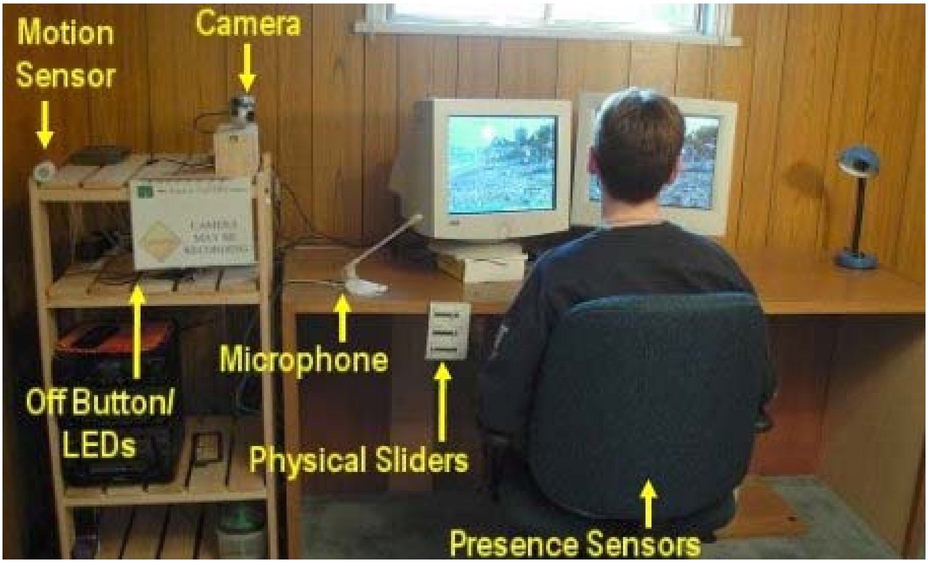


Fig. 3. The layout of the HMS within the home office/spare bedroom.

Camera State. The camera can be in one of three states: Play (Figure 1), Pause (Figures 5, 6), and Stop (Figure 7). In the play state, the camera is capturing and broadcasting video to other HMS participants (Figure 1). In the pause state, the camera no longer captures and broadcasts video to other HMS participants; however, other availability information is sent including the last video frame captured of the user and a count of the number of people in the room (Figures 5, 6). In the stop state, like the pause state, the camera no longer captures video and the last image broadcast is of the wall (Figure 7). The major difference between the pause and stop states is that it is more difficult to move out of the stop state (described in more detail later). Users can explicitly move between states by clicking the play, pause, and stop buttons (three leftmost buttons, respectively in Figures 1, 5, 6, and 7).

Capturing Angle. The camera, mounted on a rotating motor [6], is placed near the door and, given the desired camera angle, can capture any region of the room, except the doorway (Figures 3, 4: Camera). This is

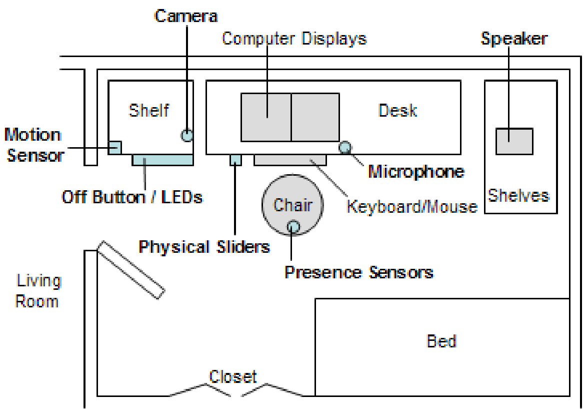


Fig. 4. An overview of the HMS layout.

Fig. 5. The HMS paused with the telecommuter leaving his chair.

Fig. 6. The HMS paused with multiple people in the room.

Fig. 7. The HMS stopped and camera facing the wall.

important as the living room is not visible (Figure 3). We provide the user with dedicated physical sliders (Figures 3, 4: Physical Sliders, Figure 8-top) and graphical sliders (Figure 2) to explicitly alter the capturing angle.

Video Fidelity. Users can adjust the captured video's fidelity by explicitly adjusting the level of blur filtration used (Figures 1, 2, 8-middle), the camera's frame rate (Figures 2, 8-bottom), or the camera's frame size (Figure 2). We provide the user with dedicated physical (Figure 3, 4: Physical Sliders) and graphical controls to explicitly adjust these three components of video fidelity. Fidelity adjustment could also be done with context-sensitive devices.

Gesture-Activated Blocking. Users can easily turn off the camera by explicitly blocking it with their hand. We detect this gesture with a proximity sensor mounted on top of the camera (Figures 3, 4: Camera, Figure 9). This can also be done using computer vision techniques [6].

Gesture-Activated Voice. Users can easily open an audio channel by explicitly moving their hand over a microphone (Figures 3, 4: Microphone, Figure 10). Moving

Fig. 8. A user adjusts the blur level with a dedicated physical slider.

Fig. 9. A user blocks the camera with his hand to turn it off.

Fig. 10. A user moves his hand over the microphone to open an audio link.

Fig. 11. A sign containing LEDs at the top and an off button.

one's hand away from the microphone closes the audio channel. We detect this gesture with a light sensor mounted on top of the microphone. This can also be done (perhaps more accurately) using other sensors, such as proximity sensors.

Easy-Off Button. Users can easily turn off the camera by touching an off button (Figures 3, 4: Off Button, Figure 11). We detect this explicit action with a capacitive sensor acting as the button, but this could also be done (and appear more realistic) with a control resembling an actual, real-world push button [17].

Telecommuter Detection. We know if the telecommuter is present at the computer by detecting (with a light sensor, Figures 3, 4: Presence Sensors) the implicit act of someone sitting down in or standing up from the desk chair. We use a radio frequency identity (RFID) tag in the pocket of the telecommuter and a RFID reader (Figures 3, 4: Presence Sensors) in the chair to identify if the individual sitting is the telecommuter. If the telecommuter is not present, we can tell how long she has been away from the computer. Our *telecommuter detection* is not a realistic solution because of limits imposed by our RFID reader, yet it works for our prototype. Other approaches could include embedding RFID tags within "work" shirts worn by the telecommuter. This helps because it can ensure the telecommuter is appropriately dressed before the HMS can be used; however, now people must wear this special garment.

Family/Friend Detection. We know if family/friends are present in the room by using an infrared motion detector (Figures 3, 4: Motion Sensor) to detect the implicit act of walking into and out of the room. This could be done more accurately with computer vision techniques; however, our solution does not require a camera to always be capturing the room's activities. Using our technique, family and friends would be more difficult to detect in rooms where a doorway is not present.

Visual Feedback. We use several visual cues to let the user know how much privacy is currently being maintained, e.g., a sign (Figure 11), LEDs (Figure 11-top), the

Table 1. Control and feedback mechanisms found in the HMS.

	1	**2**	**3**	**4**	**5**
	Attribute Controlled	**Explicit Control**	**Implicit Control**	***Audio Feedback***	***Visual Feedback***
1	Stop to Play	Click play button	None	Camera clicking; Camera rotating	LEDs on; Camera rotates to face you; Mirrored video
2	Pause to Play	Click play button	*Telecommuter sits in chair; Family/friend leaves room*	Same as above; Camera Twitches	Same as above; Camera Twitches
3	Play to Stop	Click stop button; *Block camera with hand; Touch off button*	None	Camera rotating	LEDs off; Camera rotates to face the wall; Mirrored video
4	Play to Pause	Click pause button	*Telecommuter stands up out of chair; Family/friend enters room*	Same as above	Same as above
5	Pause to Stop	Click stop button; *Block camera with hand; Touch off button*	*Telecommuter leaves the room for an extended period of time*	None	Mirrored video
6	*Capturing angle*	Adjust physical or graphical slider	Change in *camera state*	Camera rotating	Slider position; Camera position; Mirrored video
7	*Video fidelity*	Adjust physical or graphical control	None	None	Control position; Mirrored video
8	Audio link	*Moves hand over microphone base*	None	Own voice	None

camera's direction, mirrored video (Figure 1, top), and the position of physical and graphical controls.

Audio Feedback. We also use audio cues to let the user know how much privacy is currently being maintained, e.g., the sound of a camera clicking and the sound of the camera rotating [12].

There are many ways to create each of these elements and more accurate sensors exist than the ones we have chosen to use for our prototype. We have chosen methods and sensors that allowed us to rapidly and inexpensively prototype each element.

In the next section, we describe how these elements work together, along with a set of rules, to reduce privacy threats. We demonstrate this with a series of scenarios based on real telecommuting situations.

4 Rules for Balancing Privacy and Awareness in a HMS

Our HMS design uses each element within the HMS, along with a set of rules, to balance privacy and awareness for the telecommuter and others in the home. Table 1 summarizes how the design elements are either: controlled, used for explicit or implicit control, or used as feedback. Each row in the table describes how one media space attribute (column 1) is controlled either explicitly (column 2) or implicitly (column 3). The fourth and fifth columns describe the *audio* and *visual feedback* that indicate to the users that the attribute in column 1 has changed and what its current value is. The first five rows of the table describe the transitions between the three *camera states*. The remaining three rows describe other HMS attributes that can be controlled. The HMS elements, previously discussed, are *italicized* within the table.

We now present a series of scenarios that detail the privacy risks involved with using a HMS, the set of privacy rules we have created to address them, and how the HMS implements each rule to balance privacy and awareness.

4.1 Providing Awareness while Masking Embarrassing Acts

The first scenario illustrates one typical use of the HMS by a telecommuter, named Larry, who is working at home and using the media space to provide awareness to a close-working colleague at the office. Larry enters his home office/spare bedroom, dressed in casual pants and a golf shirt. While Larry is working at his computer, he suddenly sneezes. Naturally, he proceeds to blow his nose. Forgetting that the camera is capturing him, Larry begins to pick his nose at great length.

Privacy Risks: Larry is dressed appropriately to be seen at an office, yet he does not want his colleague to view him doing embarrassing, unconscious acts like picking his nose.

Rule 1: If just the telecommuter is present at the computer, the HMS assumes more awareness and less privacy is desired.

Design: This is Larry's first use of the HMS today and the *camera state* is Stop when Larry sits down at the computer. To turn the *camera state* to play (Table 1: Row 1), Larry must explicitly click the play button (Figure 1, leftmost button). Once the *telecommuter detection* has identified that it is indeed Larry at the computer, the HMS provides more awareness by moving the *capturing angle* away from the wall to record Larry. *Visual* and *audio feedback* lets Larry know the camera is now capturing (Table 1: Row 1). Larry can fine tune the awareness information and mask embarrassing acts with *video fidelity* (Table 1: Row 7).

4.2 Providing Privacy When Others Use the Computer

The second scenario illustrates what happens when the telecommuter leaves his desk and others use the computer. Larry is working at his computer when he leaves to get a coffee from the kitchen. Larry's wife, Linda, who is still in her pajamas, comes in to the home office to quickly check her email. Linda leaves the room just as Larry returns. Larry sits down and continues working.

Privacy Risks: Larry is appropriate to be viewed on camera and faces no privacy risks. Linda is not appropriate to be viewed, nor does she want to be viewed: Linda faces a threat/benefit disparity.

Rule 2: If someone other than the telecommuter is present in the room, the HMS assumes more privacy and less awareness is desired.

Design: The *telecommuter detection* knows that Larry has left his desk chair and changes the *camera state* to paused (Table 1: Row 4). *Visual* and *audio feedback* lets Larry know the camera is no longer capturing (Table 1: Row 4). The colleague maintains awareness by seeing Larry leave his chair in the last image broadcast (Figure 4).

When Linda enters the room, the *family/friend detection* flashes the LEDs and plays the sound of the camera clicking to warn Linda to make sure the camera is off. *Visual feedback* shows her that the *camera state* indeed remains paused (Table 1: Row 4). Linda checks her email and is not captured on camera.

When Larry returns to his desk chair, the *telecommuter detection* unpauses the camera, but first warns Larry this is about to happen by twitching the camera left and right (Table 1: Row 2); just as people signal their intentions, so does the camera. This complies with our third design principle. *Visual* and *audio feedback* shows Larry that the *camera state* is again Play (Table 1: Row 2).

4.3 Using Gestures to Regulate Privacy

The third scenario illustrates how the telecommuter can use gestures to control HMS attributes, which in turn affect his privacy. Larry is working at his computer composing an email and drinking his coffee. Just then, Larry knocks his mug and coffee spills all over his shirt! Larry removes his shirt and then notices the camera facing him. Larry blocks the camera with his hand then tells his colleague (through the HMS) that he has to go get a new shirt.

Privacy Risks: Larry does not want to be seen shirtless, yet he still wishes to maintain a level of awareness with his colleague.

Rule 3: The HMS must provide simple lightweight means to immediately disable the capturing device, yet still maintain awareness through alternate channels.

Design: Larry can choose one of two explicit methods to instantly stop the camera: *gesture-activated blocking* or *easy-off button* (Table 1: Row 3). *Visual* and *audio feedback* lets Larry know the *camera state* has changed (Table 1: Row 3). Larry wants to maintain awareness and tell his colleague of his predicament without using the video channel so he uses *gesture-activated voice* to open the optional audio link.

4.4 Providing Privacy When Others Enter the Room

The fourth scenario illustrates what happens when multiple people enter the home office/spare bedroom. Larry is working at his computer in the home office/spare bedroom when Linda, who has just finished taking a shower in the bathroom next door, walks into the room to retrieve her bathrobe from the closet. Linda puts on her bathrobe and leaves the room.

Privacy Risks: Linda does not want to be captured on video, especially while she is naked! Linda again faces a threat/benefit disparity, while Larry still wants to provide awareness information to his colleague.

Rule 4: If more than just the telecommuter is present in the room, the HMS assumes more privacy and less awareness is desired.

Design: The *family/friend detection* knows that Linda has entered the room and moves the *camera state* to paused (Table 1: Row 4). *Visual* and *audio feedback* indicates that the *camera state* has changed (Table 1: Row 4). Larry's colleague maintains a level of awareness with the presentation of alternate awareness information when the camera is paused: the number of people in the room, and the image of Larry sitting at his desk (Figure 5). Using these two pieces of information, it is possible for Larry's colleague to infer that Larry is still working at his desk.

Once the *family/friends detection* knows that Linda has left the room (Table 1: Row 2) and the *telecommuter detection* indicates that Larry is still at the computer, the *camera state* will return to Play once it first warns Larry with *visual* and *audio feedback* (Table 1: Row 2).

4.5 Finishing Work and Leaving the Space

The fifth scenario illustrates what happens when the telecommuter finishes working and leaves the HMS. Larry has finished working for the day and leaves the home office.

Privacy Risks: The HMS is still active when the telecommuter is finished working; future use of this room may threaten privacy.

Rule 5: If the telecommuter is away from the computer for an extended period of time, the HMS will move to a permanent, non-recording state.

Design: The *telecommuter detection* notices Larry leaving and the *camera state* pauses. After being away from his desk for five minutes, the *camera state* moves to Stop and now the last image shown to Larry's colleague is of the wall (Figure 7). This timeout interval can be customized in Figure 2. The non-recording state is permanent in the sense that to start working again, Larry must explicitly click the play button (Figure 1). Until this time, the camera will not turn on and no video will be captured; thus, no privacy violations will occur while Larry is not working.

5 Supporting Privacy Mechanisms

We now describe how we have leveraged the four categories of privacy mechanisms by designing privacy-protecting strategies for a HMS that fall into the same categories of mechanisms used by humans for privacy regulation in everyday life.

5.1 Verbal Behavior: Sound and Voice

Verbal behavior consists of the use of content and structure of what is said to control privacy [1]. For example, if a family member approaches the home office while the telecommuter is currently working she may say, "I'd like to be left alone," if she would like to have more privacy or alternatively, "please come in," if she desires interaction. We use verbal behaviors in two ways within our design: verbal instructions between media space users; and, verbal instructions or sounds cues from devices in the media space to media space users.

The first approach is trivially supported in the HMS's design for co-located HMS users (e.g., the telecommuter and others in the home): they can simply speak to others in the same location. Distance-separated users of the HMS must rely on a voice channel for this approach. The tradeoff is that we want an audio link, yet not the additional privacy threats found with a continuous audio link [16]. For this reason, our design provides an optional audio link where *gesture-activated audio* allows users to easily engage and disengage the audio link.

The second approach offers a crucial component of privacy feedback. Feedback of the level of privacy being attained is most easily presented through visuals or with audio. In the case that visuals go unnoticed, *audio feedback* becomes vital.

5.2 Non-verbal Behaviors: Presenting and Using Gestures

Non-verbal behavior consists of the use of body language, such as gestures and posture, to control privacy and can either be implicit or explicit [1]. When people are located close together, non-verbal behaviours increase [1]. For example, in an exam situation, people may try to block or cover their test paper, indicating their desire for privacy. We use non-verbal behaviours in two ways within our design: gesture-based input for devices within the media space; and, non-verbal instructions between media space users.

The first approach, gesture-based input, offers a lightweight means to control devices; users can give the media space explicit instructions using recognized hand or body motions. Our HMS uses *gesture-activated blocking* and *gesture-activated voice.*

The second approach is simply a replication of that which is done in face-to-face situations where people implicitly or explicitly use body language to control privacy. Co-located users (e.g., the telecommuter and others at home) should have little trouble with this, yet users separated by distance must rely on the video channel for presenting their non-verbal behaviors. *Video fidelity* must be high enough for other participants to easily interpret gestures and postures.

5.3 Environmental Mechanisms: Virtual Fences, Blinds, and Doors

Environmental mechanisms consist of the use of physical artifacts and features of an environment to control privacy [1]. For example, to limit neighbors from viewing one's backyard, a fence may be built or a large row of trees could be planted. Just as individuals can control their own environment in the physical world, they should be able to control their environment in a HMS. The environmental mechanisms for a HMS that we support can be grouped into three categories: lightweight mechanisms for altering the media space's physical environment; self-appropriation for controlling physical appearance and behavior; and, adjustable personal space.

The first approach allows for easy and simple privacy regulation. Our design allows explicit control over *camera state, capturing angle,* and *video fidelity*; and implicit control over *camera state* and *capturing angle.*

The second environmental approach lays in the hands of media space users. Self-appropriation involves creating an appearance and behavior suitable for the current situation [3]. Given enough *visual* and *audio feedback* of the level of privacy currently being attained, users have the power to control their own privacy by simply appropriating themselves correctly [3]. This can be difficult in a HMS however. Participants at the home location may be forced to appropriate themselves for the office, which itself can be an infringement on their autonomy. To help alleviate this problem, users can rely on lightweight controls to help users appropriate themselves correctly for both home and the office, e.g., *video fidelity.*

The third environmental approach allows HMS users to utilize personal space for controlling privacy, just like in face-to-face situations. First, the media space can be setup in any location within the home. Our HMS is setup within a home office/spare bedroom because this type of room offers users a large amount of control over their privacy because it is not commonly used by many people within the home. Second, within the media space, the camera can be positioned in any number of locations; camera placement determines what background information is captured. This typically becomes unremarkable over time [21], but care can be taken so that background information presents little privacy threat.

5.4 Cultural Mechanisms: Social Solutions

Cultural mechanisms consist of the use of cultural practices and social customs to control privacy [1]. Although it may often go unnoticed, each culture contains a set of learned social practices and customs that have evolved and developed over time [1]. We feel that in a HMS, given an established set of social protocols, users can rely on cultural mechanisms to regulate privacy when technology does not suffice. In the case that social norms are not followed, social ramifications may be in order.

6 Software and Hardware

The HMS is designed as an ActiveX® Control, which can be easily used with languages supporting Microsoft COM technologies, e.g., Visual C++, C#, Visual Basic. Two toolkits, developed in our research lab, were used to develop the HMS prototype. The first, Collabrary, makes it easy to create software with video and audio links and alter attributes such as video fidelity [7]. The second toolkit, Phidgets™, which contains pre-packaged physical devices and a corresponding software Application Programming Interface (API), makes it easy to rapidly prototype physical interfaces and sensing environments [15].

The importance of these two toolkits is that they allowed us to move our research focus away from the underlying implementation of the HMS. As such, we were able to focus our time and effort on deciding and exploring how context-aware computing could be used, what its effects would be, and if our techniques were appropriate given our research goal of balancing privacy and awareness.

7 Discussion

This was our "first cut" of a context-aware home media space and as a result we wanted to see what big problems emerged by trying it out ourselves (an evaluation methodology called "eat your own dog food"). In particular, the first author, a frequent telecommuter, routinely used the home media space over several months within his own home office/spare bedroom. We noticed several design faults. First, the control and feedback mechanisms need to be more natural if they are to fit well within a person's everyday world. For example, adjusting a physical slider to regulate privacy is a somewhat abstract notion. Second, a consequence of unobtrusive peripheral feedback of system state is that the person may overlook the information. For example, one may overlook feedback that the camera is recording at high quality (e.g., the LEDs). Third, explicit control mechanisms only work if one can anticipate and react quickly enough to the risk inherent in the current situation, if one is in a location where the system can recognize their control action (e.g., one must be near the camera to block it with a gesture); and if one feels that the effort is worth it. Ideally, the system should automatically sense privacy-violations and control the information it transmits accordingly, but this can be quite difficult to do in practice. Despite these caveats, our experience was positive overall. We were able to control the space for many situations, and indeed just knowing that we *could* control the media space was comforting. In future work, we will try to correct these problems, and we will formally evaluate the redesigned home media space as well as the general strategies we have presented.

8 Conclusion

This paper presents the rationale and prototype design of a home media space (HMS). The HMS is designed specifically for the telecommuter who chooses to work at home, but who still wishes to maintain a close-working relationship with particular colleagues at remote office environments. Our contribution is a set of five design principles for a HMS and a prototype HMS which illustrates these principles. Using these five design principles, we have created a set of privacy rules that regulate how privacy and awareness are balanced in a HMS.

Our actual use of context-aware software and dedicated physical controls has yet to be evaluated for its effectiveness in balancing privacy and awareness. However, we provide a general approach for integrating the privacy mechanisms used by people in their physical environments into a HMS. By using two toolkits, including a set of pre-packaged physical devices and sensors, we were able to focus our research on understanding how context-aware computing can be used in real-world applications. This provides a valuable contribution to context-aware computing in general.

While we have concentrated on one specific use of video in homes, our paper contributes ideas that have a broader significance for home-based videoconferencing in general. Regardless of the specific use of video in a home, people need and desire methods to regulate their privacy; many video conferencing systems (e.g., Webcam for MSN Messenger, Yahoo! Messenger) ignore these user requirements.

Acknowledgments

We would like to acknowledge NSERC and Microsoft Research for their partial funding of this research. A special thanks to Michael Boyle for the use of the Collabrary, Chester Fitchett for the use of Phidgets™, and Stacey Scott for help with editing.

References

1. Altman, I.: The Environment and Social Behavior: Privacy, Personal Space, Territory, Crowding, Wadsworth Publishing Company (1975) pp. 1-51, 194-207.
2. Altman, I., and Chemers, M.: Culture and Environment, Wadsworth Publishing Company (1980) pp. 1-12, 75-119, 155-214.
3. Bellotti, V.: Design for Privacy in Multimedia Computing and Communications Environments, in Technology and Privacy: The New Landscape, Agre and Rotenberg eds., MIT Press, (1998) pp. 63-98.
4. Bellotti, V., Back, M., Edwards, K., Grinter, R., Hnderson, A., and Lopes, C.: Making Sense of Sensing Systems: Five Questions for Designers and Researchers, Proc. CHI 2002 [CHI Letters 4(1)], ACM Press, (2002) pp. 415-422.
5. Bly, S., Harrison, S. and Irvin, S.: Media spaces: Bringing people together in a video, audio, and computing environment, Communications of the ACM 36(1), ACM Press, (1993) pp. 28-46.
6. Boyle, M., Edwards, C. and Greenberg, S.: The Effects of Filtered Video on Awareness and Privacy, Proc. CSCW'00 [CHI Letters 2(3)], ACM Press, (2000) pp. 1-10.
7. Boyle, M., and Greenberg, S.: GroupLab Collabrary: A Toolkit for Multimedia Groupware, in J. Patterson (Ed.) ACM CSCW 2002 Workshop on Network Services for Groupware, (2002).

8. Crowley, J.L., Coutaz, J., and Bérard, F.; Things That See, Communications of the ACM, ACM Press, Vol. 43, No. 3, (2000) pp. 54-64.
9. Erickson, T.: Some problems with the notion of context-aware computing, Communications of the ACM, Vol. 45(2), February 2002, (2002) pp. 102-104.
10. Fish, R.S., Kraut, R.E., and Chalfonte, B.L.: The VideoWindow System in Informal Communications, Proc. CSCW'90, (1990) pp. 1-11.
11. Fish, R.S., Kraut, R.E., Rice, R.E., and Root, R.W.: Video as a Technology for Informal Communication, Communications of the ACM, Vol. 36, No. 1, ACM Press, (1993) pp. 48-61.
12. Gaver, W. W.: Everyday listening and auditory icons. Doctoral Dissertation, University of California, San Diego (1988).
13. Greenberg, Saul: Peepholes: Low Cost Awareness of One's Community, Proc. Of CHI'96, Companion Proceedings, ACM Press, (1996) pp. 206-207.
14. Greenberg, S. and Kuzuoka, H.: Using Digital but Physical Surrogates to Mediate Awareness, Communication and Privacy in Media Space, Personal Technologies, 4(1), January (2000).
15. Greenberg, S. and Fitchett, C.: Phidgets: Easy Development of Physical Interfaces through Physical Widget, Proc. UIST 2001, ACM Press, (2001) pp. 209-218.
16. Hudson, S.E., and Smith, I.: Techniques for Addressing Fundamental Privacy and Disruption Tradeoffs in Awareness Support Systems, Proc. CSCW'96, (1996) pp. 248-257.
17. Jancke, G., Venolia, G.D., Grudin, J., Cadiz, JJ, and Gupta, A.: Linking Public Spaces: Technical and Social Issues, Proc. CHI 2001, ACM Press, (2001) pp. 530-537.
18. Kraut, R., Egido, C., and Galegher, J.: Patterns of contact and communication in scientific observation, Proc. CSCW '88, (1988) pp. 1-12.
19. Lee, A., Girgensohn, A., Schlueter, K.: NYNEX Portholes: Initial User Reactions and Redesign Implications, Group '97, ACM Press, (1997) pp. 385-394.
20. Mantei, M., Baecker, R., Sellen, A., Buxton, W., Milligan, T., and Wellman, B.: Experiences in the use of a media space, Proc. CHI '91, ACM Press, (1991) pp. 203-209.
21. Neustaedter, C.: Balancing Privacy and Awareness in Home Media Spaces, MSc Thesis, Dept. of Computer Science, University of Calgary, Canada, June (2003).
22. Schilit, B., and Themier, M.: Disseminating Active Map Information to Mobile Hosts, IEEE Network 8(5), (1994),pp. 22-32.
23. Tang, J.C., Isaacs, E., and Rua, M.: Supporting Distributed Groups with a Montage of Lightweight Interactions, Proc. CSCW '94, ACM Press, (1994) pp. 23-34.
24. Want, R., Hopper, A., Falcão, V., and Gibbons, J.: The Active Badge Location System, ACM Transactions on Information Systems, Vol. 10, No. 1, January, ACM Press, (1992) pp. 91-102.
25. Zhao, Q.A., and Stasko, J.T.: Evaluating Image Filtering Based Techniques in Media Space Applications, Proc. CSCW'98, ACM Press, (1998) pp. 11-18.

Context-Aware Computing with Sound

Anil Madhavapeddy[1], David Scott[2], and Richard Sharp[3]

[1] Computer Laboratory, University of Cambridge
avsm2@cl.cam.ac.uk
[2] Laboratory for Communication Engineering, University of Cambridge
djs55@cl.cam.ac.uk
[3] Intel Research, Cambridge
William Gates Building, 15 JJ Thomson Avenue, Cambridge, CB3 0FD, UK
richard.sharp@intel.com

Abstract. We propose *audio networking*: using ubiquitously available sound hardware (i.e. speakers, sound-cards and microphones) for low-bandwidth, wireless networking. A variety of location- and context-aware applications that use audio networking are presented including a location system, a pick-and-drop interface and a framework for embedding digital attachments in voice notes or telephone conversations.

Audio networking has a number of interesting characteristics that differentiate it from existing wireless networking technologies: (*i*) it offers fine-grained control over the range of transmission (since audio APIs allow fine-grained volume adjustment); (*ii*) walls of buildings are typically designed specifically to attenuate sound waves so one can easily contain transmission to a single room; (*iii*) it allows existing devices that record or play audio to be "brought into the user interface"; and (*iv*) it offers the potential to unify device-to-device and device-to-human communication.

1 Introduction

Researchers have spent a great deal of time studying how electronic devices can be augmented with sensors that allow them to perceive their environment. This body of work, often referred to as *context-aware* or *sentient* computing [6], stems from the idea that in order to build large-scale ubiquitous systems, individual devices must be aware of their environment (e.g. where they are, what entities are nearby, what these entities are doing, where and when something is happening). Based on such information, applications and devices can make decisions *proactively*, only engaging in interactions with humans when absolutely necessary [22].

Over the last two decades, a wide variety of sentient computing systems have been designed, built and deployed. Such systems range from complex centralised location systems that provide a high degree of accuracy [11] through to much simpler distributed tagging techniques that enable devices to detect their proximity to each other [13]. Whilst these sensor frameworks have been implemented and tested on a research-lab scale, a factor that has prevented their wide-spread global deployment is that they all rely on custom hardware. Such hardware is not

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 315–332, 2003.

usually available on the mass market and, furthermore, is sometimes expensive and difficult to install, configure and maintain.

In this paper we propose that *audio networking* can be used as the basis for developing context-aware applications. Audio networking allows standard devices fitted with speakers and microphones (e.g. PDAs, laptops, desktop PCs and mobile phones) to exchange data and infer information about their environment. One of the key advantages of audio networking is that it enables context-aware applications to be immediately deployed on a large scale without requiring users to purchase and install additional hardware.

We believe that the most important property of audio networking is *locality* since, in this respect, it has two main advantages over conventional wireless networking:

1. In the audio domain we have fine-grained control over the amplitude (volume) of the transmission and hence fine-grained control over the range of the transmission. For example, by instructing a PDA to transmit sufficiently quietly whilst holding it near a laptop one can ensure that only the laptop responds to the transmission (see Sect. 3). Conversely, one could broadcast a URL to an entire room simply by increasing the volume.
2. Walls of buildings are often *sound-proofed*; i.e. designed specifically to attenuate sound-waves in the audible spectrum. Thus, by transmitting data in the audible spectrum it is easy to infer room-grained location. Note that the same is not true for microwave radiation so, for example, it is not easy to infer location information using IEEE P802.11 [20].

We propose that the major use of audio networking is as a low-bandwidth, localised control channel. While the low-bandwidth constraint means that it is not suitable for transferring large objects, the technique is ideal for transmitting object identifiers (i.e. URLs, IP addresses etc.). Imagine a typical office environment where there are already machines connected using conventional wireless and wired networking technologies. We see audio networking providing distributed context-aware services on top of this existing infrastructure.

Putting our work into context, we observe that there is a long history of using audible signals to transmit information between electronic devices. For example, modems have been used for decades in digital communications and the telephone network uses a variety of audible signals to transfer data between handsets and exchanges [24]. More recently, researchers have revisited ideas originally exploited three decades ago in acoustically-coupled modems and devised coding-schemes that allow data to be transmitted and received through air using speakers and microphones [9]. The contributions of our research are (*i*) to extend these techniques by exploring new modulation schemes; (*ii*) to apply audio networking to location- and context-aware computing; and (*iii*) to implement applications and user interfaces that rely on audio networking.

The remainder of the paper is structured as follows: Sect. 2 describes the implementation and performance of various coding and modulation schemes that form the physical-layer of our audio networking implementation; Sect. 3, the main body of the paper, gives examples of a variety of applications that can be

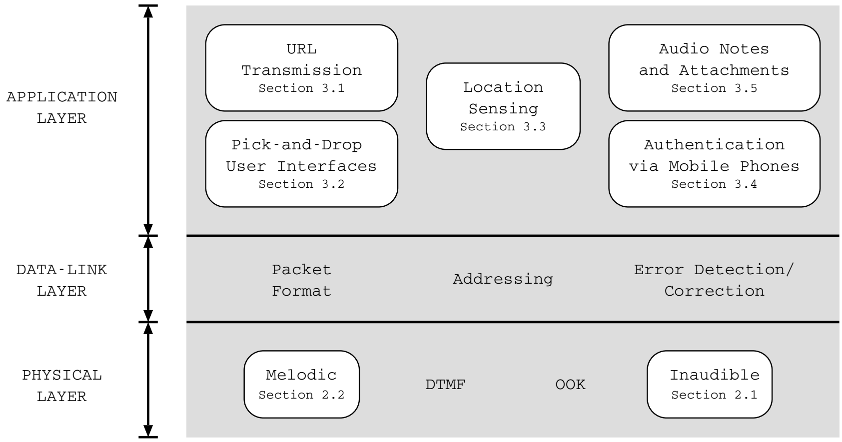


Fig. 1. A diagrammatic view of our audio networking research.

implemented using audio networking; Sect. 4 discusses related work and Sect. 5 concludes and gives directions for future work. Figure 1 presents a diagrammatic view of this paper's structure in the wider context of our audio networking research.

2 Audio Data Coding

At an abstract level the audio networking concept is very simple: transmitters modulate their data in a suitable fashion and play the resulting audio data using the host machine's speakers. Receivers listen (via their microphone) and demodulate incoming signals to recover the transmitted information.

However, although the concept itself is simple, there are a plethora of audio data modulation schemes to choose from. Each of these has different characteristics in terms of bit rate, range of transmission and, equally importantly, what the data *sounds like* to humans. The latter characteristic is particularly thought provoking as it is not one that designers of traditional wireless networking technologies have to face.

In previous work researchers have explored various coding and modulation techniques that allow data to be transmitted as audio samples [9,14]. As part of our research we have implemented four audio data transmission schemes:

1. Dual-Tone Multi-Frequency (DTMF) [24] (as used in touch-tone phones).
2. On-Off Keying (OOK) [9] modulated over a variety of audible carrier frequencies.
3. Inaudible data transmission (see Sect. 2.1).
4. Melodic data transmission (see Sect. 2.2).

Our transmission schemes divide data into fixed-size packets, each prefixed with a short preamble. A CRC field is used to detect transmission errors; invalid packets are dropped. Our audio physical layers do not guarantee delivery: as in many networking stacks, we view this as the job of higher level protocols.

Using standard sound hardware (internal laptop speakers/microphones and a pair of $10 Harman Kardon HK 206 desktop speakers) we achieved a bit rate of 20 bits/s using DTMF across a 3 meter distance with a 0.006% symbol error rate (i.e. 0.006% of DTMF tones were decoded incorrectly). We found that our OOK coding scheme was better suited to short range transmission (less than 1 meter) since at low amplitudes one does not have to worry about pulse reflections. Using OOK modulated over a 10 kHz carrier we achieved a bit rate of 251 bit/s across a 30cm distance with a bit error rate of 4.4×10^{-5}. Although these bit rates are sufficient for the applications of Sect. 3 we are confident that with further implementation effort and cleverer DSP the throughput reported here could be improved significantly.

Since DTMF and OOK are standard techniques we do not consider them further in this paper. The remainder of this section describes the technical details of our inaudible data transmission scheme (Sect. 2.1) and our melodic transmission scheme (Sect. 2.2).

The interested reader may like to hear our audio coding schemes for themselves. For this purpose, we have placed a variety of sound samples on the web [1].

2.1 Inaudible Data Transmission

Whilst audible data transmission is acceptable in many situations and sometimes even beneficial (see Sects. 3.3 and 3.5), it transpires that we can also use standard sound hardware to transmit data that is inaudible to humans.

Our current implementation of the inaudible data transmission scheme simply uses a variant of our OOK implementation, modulated over a 21.2 kHz carrier. The 21.2 kHz carrier frequency is conveniently chosen to be greater than the maximum frequency of human hearing and less than the Nyquist limit of standard sound-cards (which operate at 44.1 kHz). To ensure that users cannot hear audible clicks as the carrier switches on and off, we apply an amplitude envelope to each transmitted pulse. By increasing the attack and decay times of pulses we effectively band-limit the transmitted signal, confining it to the high-frequency (inaudible) region of the spectrum.

By analysing spectrograms of ambient noise in office environments we discovered that most noises in the 20 kHz–22 kHz region of the audio spectrum tend to occur in short bursts (e.g. doors closing, keys jangling etc.). In this environment, although our inaudible data transmission scheme is narrow-band, we can make it more robust against bursty high-frequency noise (at the cost of reducing the bit rate) by increasing transmitted pulse lengths. Our initial experiments involved broadcasting using a bit rate of 8 bits/s across a distance of 3.4 meters in an office with two occupants. The experimental setup consisted of a Dell desktop with on-board sound (Intel AC97 Audio Codec) transmitting data through Harman Kardon HK 206 speakers. The receiver was a Sony Vaio PCG-Z600LEK laptop. The occupants of the office generated ambient noise by continuing with their normal tasks (which included typing, bursts of conversation, walking around the room, moving objects and papers, answering the telephone and drinking coffee). Under these conditions, out of the 172 16-bit identifiers transmitted, 95% of them were received correctly.

Whilst 8 bits/s is undoubtedly a low transmission rate, it is still adequate for broadcasting small identifiers. For example, a throughput of 8 bits/s and a packet receipt rate of 95% is sufficient for the room-grained location beaconing example described in Sect. 3.3.

We do not have enough information to state categorically that the high-frequency inaudible transmission scheme described here will work on all sound hardware. However, we have repeated the above experiment using a variety of different microphones, speakers and sound-cards. In each case we have been able to transmit data over the 21.2 kHz carrier.

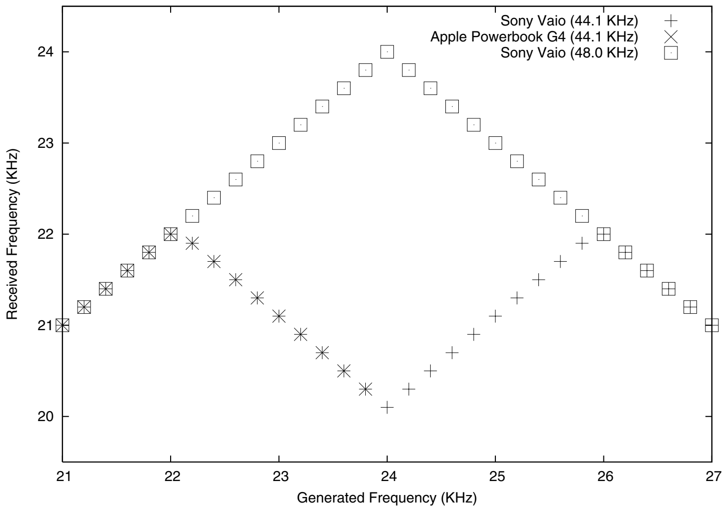


Fig. 2. A graph showing how standard sound-cards perceive frequencies above their Nyquist limit. The minima at a generated frequency of 24 kHz in the Vaio's 44.1 kHz sample-rate curve suggests, as expected, that the underlying codec is still sampling at 48 kHz.

As an interesting aside, we discovered that many sound-cards have poor or non-existent anti-aliasing filters. Experiments were performed which demonstrate that a number of sound-cards can receive sounds at significantly higher-frequencies than the Nyquist limit imposed by their sampling rate (see Fig. 2). Although frequencies greater than the Nyquist limit occur as *aliases* in the received frequency spectrum, they can nevertheless be detected and analysed. This opens the possibility of having higher frequency transmitters (e.g. computers fitted with 96 kHz sound-cards) broadcasting data in the 20-30 kHz region, which can still be received and decoded by devices with standard (44.1/48 kHz) codecs. Although our 21.2 kHz carrier was inaudible to the adults we tested it on at the amplitudes we used for transmission, increasing the separation between the threshold of human hearing and data transmission is still a desirable property. For example, young children have better high-frequency hearing than adults. By emitting data in the 24–27 kHz region and relying on the aliasing effects of standard hardware we could potentially increase transmission amplitude whilst ensuring that the signal remains inaudible to children. Investigating these ideas further is a topic of future work.

We observe that although signals modulated onto a 21.2 kHz carrier are inaudible to adult humans they are well within the hearing range of a dog. (A dog's threshold of hearing is an octave higher than that of a human: approx. 40 kHz). Further experiments are required to ascertain the effects that our inaudible data transmission scheme has on guide-dogs located in office environments.

2.2 Melodic Data Transmission

In this section we describe a framework that allows information to be transmitted as musical melodies. Although the generated melodies are unlikely to receive critical acclaim they are nevertheless "amusing little ditties" that sound surprisingly pleasant. We believe that the major application of this technique is to encode information in a form that can be played back using mobile phones (see Sect. 3.4).

Our melodic data transmission scheme assumes the existence of a set of 4 carrier frequencies. A two-bit value is signalled by playing a tone at one of these carrier frequencies for a pre-specified duration. Note that instead of activating two of these frequencies simultaneously as DTMF does, we only activate a *single* carrier frequency at a time. This enables us to encode data as *monophonic* ringtones: the base level supported by the majority of modern mobile phones.

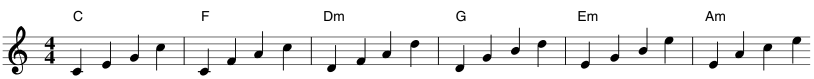


Fig. 3. Notes corresponding to carrier frequencies for first six bars of the "baroque" theme.

At any given time, we choose 4 notes from the C major (ionian) scale as our carrier frequencies. A melody is constructed by varying the carrier frequencies over time, according to a predefined sequence known both to the transmitter and the receiver. A musician is employed to choose carrier frequencies in such a way that, irrespective of the data being transmitted, the resulting melodies sound pleasing to the ear. We call a particular sequence of carrier frequencies a *theme*. Figure 3 shows an example of a theme that follows a chord sequence commonly found in baroque music. Each bar contains a group of four carrier frequencies. The nth carrier frequency (i.e. the nth crotchet) of each bar encodes the two-bit number n.

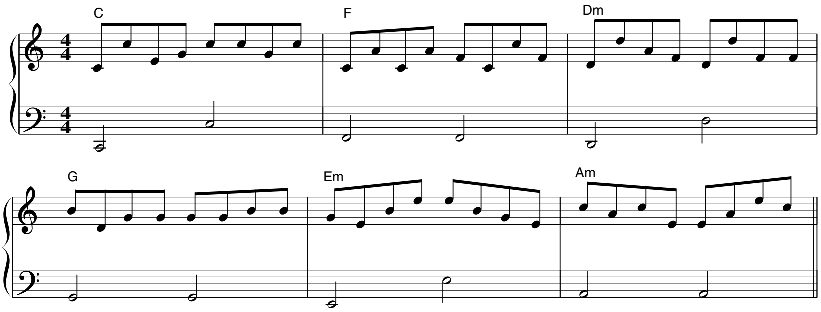


Fig. 4. An encoding of a 96-bit packet “`36FB 224D 3935 C55F 4BE4 981E`” using the *baroque* theme. (The top stave represents the data.).

Our current implementation changes carrier frequencies every 8 notes (i.e. every 16 bits). Under this scheme, the top stave of Fig. 4 shows the melody corresponding to a 96-bit packet modulated according to the “baroque-themed” frequency hopping. Two notes of the melody encode a single hex digit.

Although we use monophonic melodies to encode our information, that does not stop devices capable of generating polyphonic music from incorporating other voices in their musical output. For example, the data-transmission example of Fig. 4 is augmented with a bass-line. Although the receiver will ignore the bass-line[1] it will be heard by the human listener, making the data transmission sound more interesting.

Of course there are many themes other than the “baroque” one described above. As part of our melodic data transmission implementation, we have composed themes that represent an eclectic selection of musical styles ranging from “blues” through to “eighties pop”! We envisage a scenario where both transmitter and receiver have access to a pre-composed library of themes. The user configures their device to modulate data using their favourite theme (cf. personalisation of mobile phone ring-tones). The choice of theme is encoded as a brief preamble to the transmitted data broadcast using carrier frequencies known to both transmitter and receiver. In our current implementation we use two carrier frequencies (middle C and the C an octave above) to encode the preamble.

We have prototyped our melodic data transmission scheme using mobile phone ring-tones. Our mobile phones played the ring-tones at 7 notes per second leading to a transmission rate of 14 bits/s. We observed that although both receiver and phones are capable of faster transmission[2], increasing the tempo of the ring-tones beyond 8 or 9 notes per second makes them sound less pleasant to human listeners: the ear does not have enough time to pick out the melody.

[1] It will be filtered out since it occurs outside the frequency range in which data is expected (and is thus treated as unwanted noise).

[2] We achieved 83 bits/s (the fastest that the mobile phone could play ring-tones) with an upper limit of 0.001% bit error rate; the phone was held 2cm from the microphone.

Since the melodic nature of the data encoding is a feature we wish to emphasise, we see 14 bits/s as the upper limit of this encoding technique. This is more than adequate for transmitting short (e.g. 96-bit) IDs (see Sect. 3.4).

3 Audio Networking: Example Applications

In Sect. 1 we motivated audio networking as a "low-bandwidth localised control channel". Here we expand this idea further, giving concrete examples of applications based on audio networking technology.

The case studies presented here are designed to illustrate different properties of audio networking. Sections 3.1 and 3.2 use audio for local transmission of object identifiers (e.g. URLs); Sect. 3.3 demonstrates the room-grained locality property of audio networking; and Sects. 3.4 and 3.5 show how existing devices that can record or play sounds can be "brought into the interface".

3.1 Local Transmission of URLs

To demonstrate the locality properties of audio networking we implemented a "shared collaborative environment" that allows devices in close proximity to each other to transmit and receive URLs in the spirit of HP's CoolTown project [12]. In our current implementation, received URLs are automatically added to a pre-specified bookmarks folder, allowing users to peruse them at their leisure.

Although transmitting short URLs is fine, we soon found that it can take an unacceptably long time to transmit long URLs over low bit-rate audio channels (e.g. consider a complex dynamic web application with a large number of HTTP-GET parameters). To alleviate this problem we introduce another layer of indirection via an *object server*, a networked machine that associates short *object IDs* (i.e. primary keys) with URLs. Recall that, in Sect. 1, we motivated audio networking as a control-channel for existing high bandwidth networks. By using an object server we can design protocols that transfer as much as possible using the existing wired/wireless networking infrastructure whilst still exploiting the locality properties of sound.

In this framework, our local shared collaborative environment works as follows: the transmitter (*i*) creates a new record on the object server that contains the URL to be transmitted (using the high-bandwidth network); and (*ii*) broadcasts the corresponding unique identifier over audio. Receivers decode the identifier and fetch the corresponding URL from the object server. Essentially object IDs are just *relative URLs* which, although not globally unique, are unique to a particular object server. Of course, we can make object IDs globally unique by prefixing them with the IP address of the object server on which they are stored. However, since our aim is to make object IDs as small as possible, we assume for now that both transmitter and receiver are configured to use the same object server. This setup is ideal for transmitting objects around an office-scale environment.

The object server implementation is straightforward, comprising a webserver, a SQL database and scripts that deal with insertion and retrieval of

URLs (over standard HTTP). A number of housekeeping tasks are performed automatically according to a configurable system-wide policy. For example, entries that time-out are automatically removed. This allows us to keep the pool of active objects small, reducing the lengths of object identifiers. In contrast to URLs, which are often designed to be persistent, object IDs frequently only exist for a few seconds: the time it takes devices to receive them over audio and dereference them over a high-bandwidth network.

3.2 A Pick-and-Drop User Interface

A *Pick-and-Drop user interface* [18] enables users to move virtual objects around in physical space by associating them with physical devices. We have implemented an interface that allows users to *pick up* virtual objects (e.g. files, documents) onto their PDAs, carry them around a building (using the physical PDA to represent the virtual object) and *put them down* on other devices. As a motivating example for this kind of user interface, consider a person who picks up a document from their desktop machine, walks across a building and puts it down on a printer (at which point the document is automatically printed)[3].

User interfaces such as this have been studied at great length. What makes our interface interesting is that it is implemented using audio networking. When an object is *picked up*, the transmitting device copies it to a local web-server, inserts the object's URL into the object server and transmits the corresponding object ID to the PDA over audio. Similarly, when an object is *put down*, the PDA broadcasts the object ID to the receiving machine over audio, which fetches the corresponding document and acts accordingly. We believe that there are two main advantages to this approach: (*i*) it is built out of existing, commonly deployed hardware; and (*ii*) giving the user fine-grained control over the transmission range (i.e. volume) gives the user an extra degree of flexibility. For example, transmitting the object ID over at an amplitude corresponding to a 4-meter range may copy the corresponding object to all computers in a room; conversely transmitting over a 1 cm range would only transmit the object ID to the single device held next to the PDA.

We have implemented a simple GUI for transmitting files using object IDs (see Fig. 5). On the transmitter side, the user drags files into the *transmit window* (at which point the document is copied to the object server, an ID is assigned and broadcast over audio to a local PDA). On the receiver side the interface consists of a menu of recently received objects. By selecting an object, one can open it on the PDA (if the device supports the file-type) or retransmit the object ID as audio. A configuration dialog allows the interface to be customised to suit the user: e.g. whether objects should automatically be removed from the menu once they have been transmitted, or over what range the object IDs should be broadcast (see Sect. 3).

[3] Since printers don't have microphones we would probably have to place a device near the printer that receives object IDs and issues print commands. However, since in many cases there are print servers located next to printers, these devices could often be used for this purpose.

Fig. 5. *Left:* The GUI interface for receiving/transmitting object IDs; *Right:* Transmitting a document ID for printing using audio networking.

Although machines that access the objects themselves must be connected to high-bandwidth networks, one can still move object IDs around using devices without networking capability. However, there is one subtlety which arises if the PDA shown in Fig. 5 is not networked: in order for users to browse the object IDs currently held on the PDA it is essential that human-readable names (e.g. filenames) are associated with object IDs. When a device receiving an object ID has access to conventional higher bandwidth networking then they can use this medium to query the object server, obtaining the object name from the ID. In this case only the object ID itself has to be broadcast over audio. However, when a device receiving an object ID does not have access to higher bandwidth networking then the object name must be transmitted over audio along with the corresponding ID.

Short range transmission. Although room-scale multicast is a useful property of audio networking, there are a number of scenarios where very short range transmission (i.e. 0–2 cm) is appropriate (see above). Short range audio transmission has a number of useful properties. In particular:

1. Transmission amplitude can be reduced so as not to cause distraction to others in the same environment. We have demonstrated that by holding the device within approximately 2 cm of a receiver's microphone we can transmit data at comparable volume to playing quiet music through headphones or listening to a voice message through a telephone handset.
2. In a room that contains several devices, short range transmission provides a simple and intuitive mechanism for a user to select a single device to communicate with. Devices only communicate if they are less than a few centimeters apart.

3.3 Room Grained Location

In Sect. 1 we observed that, since buildings are specifically designed with soundproofing in mind, sound tends to be attenuated by walls. As a result, audio signals provide a convenient mechanism to infer room-grained location. Figure 6 shows experimental evidence that justifies this claim. We measured the received amplitude of a 21.2 kHz pulse (transmitted from a pair of desktop speakers) on both sides of an office wall, using the built-in microphone and sound-card on a Sony Vaio laptop. (Recall that 21.2 kHz is the carrier frequency used for our inaudible transmission scheme.) Although inside the room the tone is loud, outside the room it is not detectable.

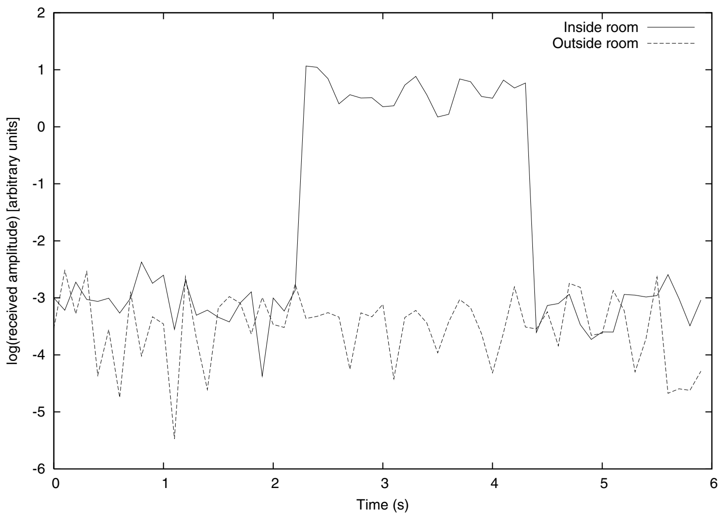


Fig. 6. A graph showing the received amplitude of a 2 second 21.2 kHz pulse on the both sides of an office wall 3 meters away from the speaker. Outside the room the tone is barely detectable: attenuation of > 20 dB.

The simplest audio location technique involves placing a *beacon* in each room that repeatedly broadcasts a unique *room identifier*. Devices placed in the same room as the beacon decode the room identifier and use this information to infer their location. We do not advocate the use of extra hardware for room beacons. Instead, we note that in many office environments there is at least one device in each room that seldom moves (e.g. a desktop computer). A simple *beaconing service* (which repeatedly plays a sound sample containing the room id) is installed on this machine.

The major concern with this method is that the constant beaconing may annoy users. To alleviate this problem we employ our inaudible transmission scheme (see Sect. 2.1) in the beaconing service. Our prototype implementation transmits a room identifier every 15 seconds; room identifiers take 3.5 seconds to be transmitted. To demonstrate the utility of room-grained location we have implemented an application that automatically configures a laptop to use its nearest printer as it is moved around a building.

Whilst the beaconing technique allows devices to infer their own location, it does not provide a mechanism to query the location of other devices. To address this limitation we implemented another system in which fixed desktop machines run *listener services* that detect short (0.3 s) 800 Hz tones and report back to a centralised *location server*. We assume that the devices whose location we want to query already have conventional (wired or wireless) networking capability. In this framework a location query can be performed as follows: (*i*) the device to be located is signalled over the network; (*ii*) the device modulates a short code over a 800 Hz tone; and (*iii*) any listener services that hear the tone report back to the location server which aggregates the information in order to infer which room the device is in.

Instead of the location server continually polling devices, the system is intended to query the location of a device only in response to a user's request. In this context, we deliberately choose an audible signal (800 Hz) for the device's "I'm here" message, since it performs the useful function of notifying nearby humans that the location of a device has been queried. It is interesting to note that in ORL's Active Badge System [25] a speaker on the badge was specifically employed to notify the wearer when their location had been queried. Stajano argues that this *principle of reciprocity* makes the location-system more palatable to users worried about loss of privacy [21]. A nice feature of our implementation is that we are able to unify the signalling of the listening service and the notification of the human observer in a single audible broadcast.

There are a number of other audio location architectures which one can envisage. For example, whereas the location system proposed above signals the device to be located over an existing (wired or wireless) network, one can imagine a system that signals the queried device using an audio broadcast played simultaneously in all rooms. This avoids the requirement that locatable devices must be connected to an existing network. Another variant of this idea is to modify the listening service to detect mobile phone ring-tones. In this context, mobile phones (and therefore, with high probability, their owners) can be located. In our experiments with this technique we configured our mobile phones to play a single-note ring-tone when called from a known phone used to initiate location queries. In this way one can easily disambiguate location queries from genuine phone calls. (The *caller groups* feature, which allows specific ring-tones to be associated with specific callers, is a common feature on modern mobile handsets.)

3.4 Authentication Using Mobile Phones

Section 2.2 shows how data can be encoded as melodies. Using this encoding scheme, unique identifiers can be sent to users' mobile phones as ring-tones. An interesting application of this technology is to use ring-tones as capabilities: by playing back a received ring-tone users can authenticate themselves to a device.

As a real-world application of this technique, consider an automatic ticket-collection machine (of the type commonly found in cinemas) that prints tickets that have been pre-booked over the Internet or telephone. Systems of this type typically use credit card numbers to authenticate users: when a credit card is inserted into the collection machine, the appropriate tickets are generated. However, since children tend not to own credit cards (and may not be trusted to borrow their parents' cards) they are precluded from using the ticket-collection device.

Consider the scenario where a parent purchases cinema tickets for a group of 14 year-old children[4]. As part of the booking system the parent enters the mobile phone number of one of the children. An authentication ring-tone is sent to this phone via SMS, allowing the children to collect the tickets without having to borrow the adult's credit card.

Using ring-tones to transmit data has different properties to transmitting data using Bluetooth [2], which is increasingly becoming integrated into mobile phones. In particular, whereas Bluetooth offers significantly higher bandwidth, audio supports stricter locality properties. These properties complement each other well: consider a scenario where a mobile phone uses an audible exchange to authenticate itself against a local Bluetooth device. Once audio has been used to introduce the devices, higher-bandwidth communication can be initiated automatically over Bluetooth. This removes the current requirement for the user to enter an authentication PIN into the Bluetooth device they wish to communicate with.

We believe that the mobile-phone audio authentication technique described here will become more flexible as the Multimedia Message Service (MMS) continues to gain acceptance over SMS [7]. Using MMS one can transmit large audio samples to mobile phones.

3.5 Audio Notes and Attachments

Audio networking allows devices that do not currently inter-operate with computers, to be "brought into the interface". In Sect. 3.2 we have seen how object IDs can be recorded on PDAs to create a user interface where users can physically move objects around a building. However, we do not have to rely on PDAs to record object IDs: any device that can record audio is sufficient. In this section we present two applications that use existing recording devices and transmit object IDs. In both cases, the transmitter uses our drag-and-drop interface to copy object URLs to the object-server and transmit unique audio IDs (see Sect. 3.1).

By recording object IDs using standard voice recorders (i.e. cellphones, analog dictaphones, voice mail systems etc.) we can *attach* digital documents to voice notes. For example consider the memo: "Dave sent me revision 2 of this document today $\langle$insert object $ID_1\rangle$; must forward it to Anil $\langle$insert object $ID_2\rangle$" where ID_1

[4] We conveniently choose an age at which children are (*i*) old enough to go to the cinema by themselves; (*ii*) not old enough to own credit cards; and (*iii*) likely to own a mobile phone!

corresponds to a Word Document and ID_2 corresponds to Anil's email address (i.e. a `mailto:` URL). When the object IDs are played back near a networked computer it can receive the IDs, fetch the associated URLs and act accordingly (e.g. open the Word Document and open a blank email window with the `To:` field containing Anil's email address).

Another similar example involves embedding objects IDs in telephone conversations. As part of a conversation the transmitter tells the receiver that they would like to transmit an object. At this point, the transmitter holds the handset to their computer's speaker and drags the document to the transmit window (resulting in an audible object ID); the receiver holds the handset to their computer's microphone to receive the object ID[5]. As long as both computers are configured to use the same object server, the file simply appears on the receiver's computer. Both users then pick up their handsets again and continue their conversation. Whilst the idea of using acoustic-coupled software-modems as an out-of-band channel to control higher bandwidth networking infrastructure is technically straightforward, we have found the resulting applications to be useful and intuitive. Just as documents can already be attached to emails, audio networking allows files to be attached to telephone conversations and voice notes.

DTMF is a suitable transmission mechanism for these applications since it uses the voice frequency-band and therefore performs well over devices such as telephones and dictaphones. Our current implementation of these applications uses fairly low-speed DTMF (4 tones or 16 bits/s) to transmit object IDs. The main reason for slow transmission in these examples is that object IDs are recorded and replayed a number of times in the analog domain. We have to make sure that transmitted data is robust against the signal degradation that occurs as a result. Of course, 16 bits/s is perfectly adequate for transmitting short object IDs.

4 Related Work

Many devices exist that offer wireless data transmission capabilities. Amongst the most popular, IEEE 802.11b [8] and Bluetooth [2] use relatively low power radio transmissions in the 2.4 GHz band to provide local area networking. On a smaller scale, Motes are low-power sensor devices that have been successfully deployed in an environmental monitoring project [15]. Motes use a short range radio link to communicate sensor readings with each other and with a base station. The PEN [5] project (previously named "piconet") focused on using very infrequent, low power radio transmissions to create a low-bandwidth localised control channel.

There are a number of factors that differentiate our work from these technologies. Firstly, our "audio network interface cards" are ubiquitous – after all, most

[5] Of course one can envisage a more developed framework where computers are given a direct audio interface to the user's telephone. However, since this kind of interface is not considered standard hardware it is not the focus of this paper.

computers and PDAs, as well as other common devices such as mobile phones and dictaphones, come equipped with the ability to record and play back sound. Secondly, many existing microphones are designed to be unidirectional whilst, in contrast, radio antennae are typically omni-directional. As a result audio devices must be roughly pointing at each other in order for communication to take place. We see this as an advantage: it makes it more difficult for devices to communicate by accident. (However, note that audio is not as directional as IrDA). Thirdly, we directly benefit from the low-level nature of the sound interface. Whereas a network card typically offers a fixed set of APIs, the sound interface is just a DAC. Since all signal processing must be done in software we are able to directly manipulate the representation of packets at the sample level, gaining the same kind of flexibility as software radios [23]. One major advantage is that we can easily implement and test new application-specific features. For example we added a *send packet quietly* API to transmit a packet over a shorter range by reducing the amplitude of the generated samples. A further benefit is that we can infer more information from each transmission than just a packet's payload. For example, the received amplitude of the signal is an indication of distance between sender and receiver.

There are a wide variety of location systems described in the literature. Notable examples include active bats [26], active badges [25] and crickets [17]. Unlike our audio-based system, these projects all require custom hardware. The RaDaR [3] project, which attempts to infer location from the signal strength of an IEEE P802.11 WaveLAN interface, is closer to our approach. However, it suffers from two critical problems: (*i*) it is limited to using only the APIs that come with the WaveLAN card (i.e. the software does not have access to the raw RF signal); and (*ii*) WaveLAN is designed to permeate buildings in order to provide network coverage. Since, in contrast, buildings are explicitly designed to *block* audio transmissions we argue that audio is a more appropriate medium for inferring room-level location.

Although a lot of work has gone into the design of human-to-computer audio interfaces [19] (and conversely computer-to-human audio interfaces [16]), the idea of using audio for computers to communicate with each other seems to have been largely overlooked in mainstream computing. We believe that this may be, at least partly, due to a concern that audio transmission is too intrusive to humans. We hope that the applications presented here (in particular, see Sects. 3.3 and 3) demonstrate that this is not the case.

Garasimov and Bender [9] have investigated a number of techniques for transmitting information in the audible spectrum. They concentrated mostly on increasing bandwidth e.g. fast On-Off Keying (OOK) and Code Division Multiple Access (CDMA). The bandwidths they report are high (3.4 Kbits/s for CDMA) although no error rates are presented. Their paper states that it is difficult to do inaudible data transmission using standard hardware, claiming that standard 44.1 kHz DACs do not work well enough to transmit data over frequencies greater than 18.4 kHz (a frequency that is audible to most people). In contrast, we found that the DACs on cheap sound-cards worked well at much higher frequencies (even above the Nyquist limit due to lack of anti-aliasing filters, see Sect. 2.1). The bit-rates we achieved whilst modulating data over a a 21.2 kHz carrier, although low, are sufficient for a variety of context- and location-aware applications (see Sect. 3.3).

Lopes and Aguiar have also studied mechanisms for encoding data melodically [14]. The main contribution of our work in this area is the application of melodic data encoding to mobile phone ring-tones. Also, our scheme is the first to allow data to be *themed* in the musical style preferred by a user. The existence of a multi-million-dollar industry in themed ring-tones suggests that this is an important consideration!

The use of audio communication has interesting security properties. Balfanz et al [4] considers the use of audio as a "privileged side channel" to bootstrap trust between strangers. His proposed system relies on the property that although the side channel can probably be snooped (e.g. case using a very sensitive microphone) it is not possible for attackers to subvert the protocol without drawing attention to themselves (e.g. by making a loud noise). The implementation and deployment of such a system using audio networking technology is an interesting area of future work.

5 Conclusions and Further Work

In this paper we have motivated audio networking as a low-bandwidth localised control channel and described the implementation and performance of a variety of audio data transmission schemes. A number of applications based on audio networking have been described demonstrating that location- and context-aware applications can be based on existing deployed hardware using audio networking techniques.

In future work there are a number of other audio transmission mechanisms that we are keen to explore. For example, by mixing our high-frequency narrow-band transmission scheme (see Sect. 2.1) with band-limited samples in the audible spectrum we are able to combine inaudible data transmission with audible noises. A possible use for this technique is to accompany inaudible transmitted data with an audible sound, the sole purpose of which is to provide information to human listeners. For example, an object ID (see Sect. 3.1) broadcast in the 21 kHz band could be augmented with a sample that speaks the name of the object (e.g. via voice synthesis) in the 0–10 kHz band.

Digital watermarking research has explored a number of ways in which inaudible data can be hidden in audible sounds. A technique that we find particularly interesting is hiding data in music [10]. We are keen to ascertain whether we could use this technique to hide data in background music played in department stores, restaurants etc. Note also that hiding data in other audio-streams would allow us to use audio networking techniques in environments where people are already using their PC's speakers for other purposes. This would be extremely useful in office environments, for example, where people often use their desktop speakers to play music.

One of the advantages of audio networking is that the flexible APIs allow one to infer more information from a received packet than just its payload data (see Sect. 4). In future work we intend to explore this idea further. For example, how accurately can one measure distances between devices using an audible "ping"?

Audio networking has a variety of interesting properties that differentiate it from conventional wireless networking technologies:

1. the software has fine-grained control over the audio physical layer;
2. building sound-proofing constrains transmission to a single room; and
3. device-to-device and device-to-human communication can be unified.

Furthermore, since any device that can play or record sound can take part in audio networking, the technique relies entirely on ubiquitously available hardware. We hope that the research reported in this paper paves the way for location- and context-sensitive applications that can be deployed immediately using existing infrastructure.

Acknowledgements

This work was supported by Network Appliance, Inc., Intel Research, and the Schiff Foundation. The authors wish to thank Derek Mcauley, Alistair Beresford, Andrew Moore, Dorothy Moore, Madeleine Glick, Charles Berlin, Mark Hagaard, Jon Crowcroft, Alan Mycroft, James Scott, Tim Harris, Frank Hoffmann and the anonymous reviewers for their valuable comments and suggestions.

References

1. Audio Networking Example Sounds: `http://www.recoil.org/audio/`.
2. The Bluetooth Specification version 1.1. `http://www.bluetooth.com/`.
3. P. Bahl. User location and tracking in an in-building radio network. Technical Report MSR-TR-99-12, Microsoft Research, February 1999.
4. D. Balfanz, D. Smetters, P. Stewart, and H. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Network and Distributed Systems Security Symposium*. Internet Society, 2002.
5. F. Bennett, D. Clarke, J. Evans, A. Hopper, A. Jones, and D. Leask. Piconet: Embedded Mobile Networking. *IEEE Personal Communications*, 4(5):8–15, 1997.
6. Diego Lopez de Ipina and Sai Lai Lo. Sentient computing for everyone. In *Distributed Applications and Interoperable Systems*, pages 41–54, 2001.
7. ETSI. Universal Mobile Telecommunications System (UMTS); Multimedia Messaging Service (MMS); stage 1. TS 122 140 V5.4.0 (2002-12) - (3GPP TS 22.140 version 5.4.0 Release 5).
8. IEEE Standards for Information Technology. *Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Network - Part 11: Wireless LAN Medium Access Control and Physical Layer Specifications.* 1999.
9. V. Gerasimov and W. Bender. Things that talk: Using sound for device-to-device and device-to-human communication. *IBM Systems Journal*, 39(3 and 4), 2000.

10. Daniel Gruhl, Anthony Lu, and Walter Bender. Echo hiding. In *Information Hiding*, pages 293–315, 1996.
11. Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. In *Mobile Computing and Networking*, pages 59–68, 1999.
12. Tim Kindberg and John Barton. A Web-based nomadic computing system. *Computer Networks (Amsterdam, Netherlands: 1999)*, 35(4):443–456, 2001.
13. D. Kirsch and T. Starner. The Locust Swarm: An environmentally-powered, networkless location and messaging system. In *1st International Sumposium on Wearable Computers*, pages 169–, 1997.
14. Cristina Videira Lopes and Pedo M.Q. Aguiar. Aerial acoustic communications. In *International Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, October 2001.
15. Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless sensor networks for habitat monitoring. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta, GA, September 2002.
16. Elizabeth D. Mynatt, Maribeth Back, Roy Want, Michael Baer, and Jason B. Ellis. Designing Audio Aura. In *Conference on Human-Computer Interaction*, pages 566–573. ACM, 1998.
17. Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket location-support system. In *Mobile Computing and Networking*, pages 32–43, 2000.
18. Jun Rekimoto. Pick-and-drop: A direct manipulation technique for multiple computer environments. In *Proceedings of User Interface Software Tools (UIST)*, pages 31–39, 1997.
19. Alexander I. Rudnicky, Alexander G. Hauptmann, and Kai-Fu Lee. Survey of current speech technology. *Communications of the ACM*, 37(3):52–57, 1994.
20. Siddhartha Saha, Kamalika Chaudhuri, Dheeraj Sanghi, and Pravin Bhagwat. Location determination of a mobile device using IEEE 802.11 access point signals. Master's thesis, Indian Institute of Technology Kanpur, 2002.
21. Frank Stajano. *Security for Ubiquitous Computing*. John Wiley and Sons, February 2002.
22. David Tennenhouse. Proactive computing. *Communications of the ACM*, 43(5):43–50, 2000.
23. David L. Tennenhouse and Vanu G. Bose. Spectrumware: A software-oriented approach to wireless signal processing. In *Proceedings of The First International Conference on Mobile Computing and Networking*, pages 37–47. ACM, 1995.
24. International Telecommunication Union, editor. *CCITT Blue Book Recommendation Q.23: Technical Features of Push Button Telephone Sets*. Geneva, 1989.
25. Roy Want, Andy Hopper, Veronica Falcao, and Jon Gibbons. The Active Badge Location System. Technical Report 92.1, ORL, 24a Trumpington Street, Cambridge CB2 1QA, 1992.
26. A. Ward. *Sensor-driven Computing*. PhD thesis, University of Cambridge, August 1998.

An Architecture and Framework for Steerable Interface Systems

Anthony Levas, Claudio Pinhanez, Gopal Pingali, Rick Kjeldsen, Mark Podlaseck, and Noi Sukaviriya

IBM T.J. Watson Research Center
Hawthorne, New York, USA
`http://www.research.ibm.com/ed/`

Abstract. Steerable Interfaces are emerging as a new paradigm used in realizing the vision of embodied interaction in ubiquitous computing environments. Such interfaces steer relevant input and output capabilities around space, to serve the user when and where they are needed. We present an architecture and a programming framework that enable the development of Steerable Interface applications. The distributed multi-layer architecture provides applications with abstractions to services of several novel components – for instance, steerable projection, steerable visual interaction detection, and geometric reasoning. The programming framework facilitates integration of the various services while hiding the complexity of sequencing and synchronizing the underlying components.

1 Introduction

Weiser [1] characterizes "good" technology as being invisible and highlights the trend in ubiquitous computing towards "invisible" computers. The quote, "Invisible technology stays out of the way of the task - like a good pencil stays out of the way of writing", stresses the importance of technology allowing the user to focus on the task at hand, and allowing them to take advantage of tacit and contextual knowledge, while unencumbered by the technology itself. As a principle for building invisible interfaces, he remarked, "the unit of design should be social people in their environment, plus your device". As the focus of interaction moves away from the desktop focusing on the individual, his relationship to the physical world and the objects that inhabit it, new technologies will be needed to afford effective interaction.

Dourish [2] has elaborated the notion of *embodiment*[1], and argues that Tangible [3] and Social Computing [2] are aspects of the same perspective he calls Embodied Interaction, which we believe forms a compelling basis for studying and building interactions in ubiquitous environments. In his view "Embodied interaction provides conceptual tools for understanding how the interface might move into the background

[1] "embodiment …denotes a form of participative status, …and is about the fact that things are embedded in the world and the ways in which their reality depends on being embedded."

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 333–348, 2003.

without disappearing altogether." [2]. Rehman et al. [4] argue that the trend towards invisible computers have left the user without meaningful cognitive models of system function and behavior. They propose a solution can be found in the use of the "Many, many [input/output] displays" Mark Weiser [1] predicted would be available in ubiquitous environments of the future. Rehman addressed this problem using head mounted displays and cameras to present information to users in a contextually appropriate manner. We share the common view expressed by Dourish and Rehman that it is important to afford natural interaction to the user when and where they need it in the environment without disappearing altogether. However, we feel that wiring the environment with many displays is difficult and that wearing head mounted gear or carrying around devices is encumbering. Our work focuses on a new paradigm we are developing called Steerable Interfaces (SI) [5]. The main principle here is to use devices that can steer relevant input and output capabilities to the user in the environment, when and where they are needed. This obviates the need to carry or wear devices or to outfit the environment with many displays or other devices.

Steerable Interface technology enables the computer interface to dynamically reach into the environment and interact with the user based on where he is and what he needs. For example, an interactive display may be projected onto a nearby table that allows the user to manipulate a projected control (e.g. slider) to change some feature of the environment. Physical objects may be dynamically annotated with projected information regarding their use and endowed with behavior through steerable input provided by visual gesture recognition devices. This would allow the user to manipulate them in a tangible manner to achieve a desired goal. Instead of having many interactive displays of varying sizes, projected displays may be resized to suit the needs of the interaction, perhaps small for private notification or large for interacting with colleagues. A projection could also concurrently simulate many interactive displays on a given surface by partitioning its overall space into smaller displays that have the appearance of being separate.

The computer no longer needs to be thought of as a device that is viewed through a fixed screen found in a static location, but as a device that can "reach" into our "real" world to interact with us. One may view this as embodying the computer such that it has a dynamic presence in our world and can interact with us in a goal directed manner. Nothing, however, prevents us from creating "bad" interfaces using this technology other than grounding our work in principles developed through the study of Embodied Interaction.

Pinhanez [6] describes the Everywhere Displays (ED) Projector, a projector with an associated rotating mirror that enables images to be projected onto virtually any surface while correcting for oblique projection distortion. Kjeldsen et al. [7,11] discuss adding Steerable Interaction capabilities to the ED, through a steerable vision system that can recognize hand gestures. Pinhanez et al. [8] and Sukaviriya et al. [9,10] describe several applications that highlight the use and advantages of SI's. Pingali et al.[5] have defined the concept of SI and described its characteristics and articulated the enabling technologies. This extends the use to the acoustic domain through the use of steerable microphone arrays [12] for input and steerable audio output [13]. Lai et al., [14] have developed a prototype of an adaptive, personalizable workspace, called BlueSpace, using this technology. This workspace application has received significant publicity in the media to date.

This paper focuses on an architecture and programming framework that supports the development of Steerable Interfaces. This paper is organized as follows. Section 2 describes the requirements that drove the architectural design as well as the resulting architecture. Section 3 discusses the requirements and design of the software development framework. Section 4 illustrates the use of the Java Application Development Environment and highlights productivity gains. Section 5 demonstrates system use through an example application. Section 6 concludes with a discussion on what we have learned and directions for future work.

2 System Architecture

Our primary goal here is the development of an architecture that facilitates the development of effective tools and techniques for building and delivering Steerable Interfaces, as well as the exploration of new modes of interaction for ubiquitous computing environments. This ranges from the design and development of specific technologies, for example in projection and visual interaction, as well as the combination of technologies to form new interaction capabilities.

Analysis of our primary goal resulted in the following set of requirements for an architecture that enables Steerable Interfaces:

- Support communication among the multiple hardware and software components used in Steerable Interface systems.
- Insure that components support the definition of an interaction that is independent of the location where the interaction is delivered. Allow the same interaction to be delivered to any available location in the environment.
- Provide a clear abstraction between function and implementation of components to facilitate experimenting with new ways to deliver underlying function without affecting applications that use the component.
- Insure components operate independently as well as work together.
- Facilitate the easy addition of new components to extend functionality.
- Afford both asynchronous and synchronous operation of components from layers above.
- Enable development using different languages and development approaches.
- Support a distributed architecture.
- Insure adequate performance.

Figure 1 shows our architecture for Steerable Interfaces. It is a three-tier architecture composed of a Services Layer, an Integration Layer and an Application Logic Layer, as explained below.

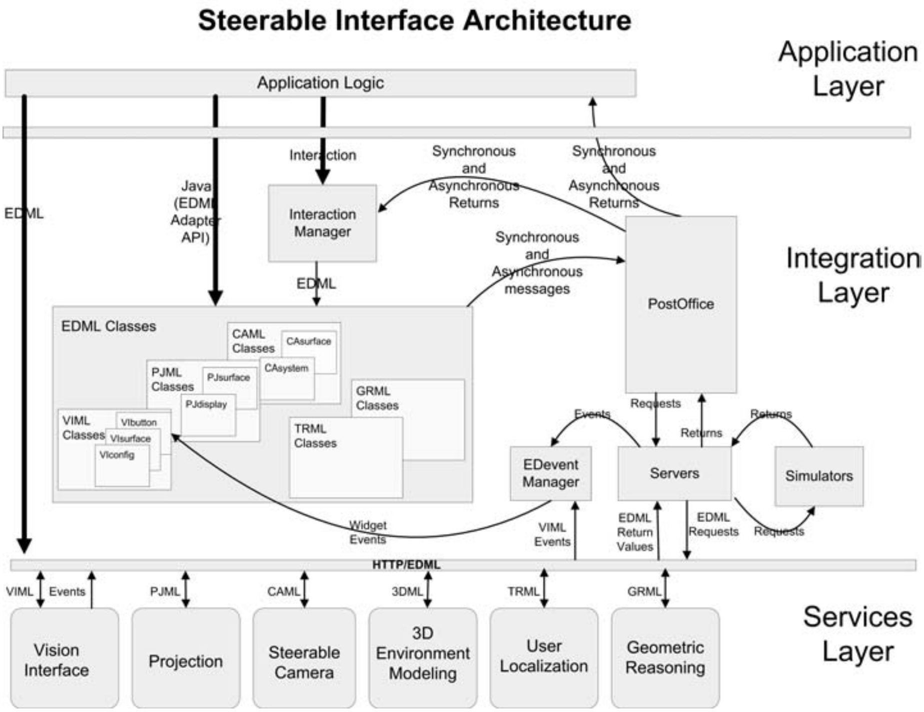


Fig. 1. An architecture for steerable interfaces.

2.1 The Services Layer

The Services layer consists of a set of modules that represent independent components of a Steerable Interface system. Each of these modules exposes a set of core capabilities through an HTTP/XML Application Programming Interface (API). Modules in the Services Layer have no "direct" knowledge or dependence on other modules in that layer. The modules share a common XML language called the Everywhere Displays Markup Language (EDML) along with a specialized dialect for communication with each module in this layer. Applications written in any language can directly communicate with each module in this layer using EDML, as indicated by the first arrow on the left in Fig. 1, labeled EDML. EDML allows us to explore developing applications using a variety of different approaches by providing language independence. Multiple modules of each type can be instantiated and operated at the same time.

The core EDML definition includes commands for establishing communication with a module, starting and stopping their respective services, and commanding or querying modules. Basic EDML action commands fall into 3 logical pairs: Use/Release are used for definition/allocation and de-allocation of objects (e.g. Buttons, Images, etc.). Set/Get are used for setting or retrieving values of objects and Activate/Deactivate for activation and deactivation of "Used" (allocated) objects.

All modules in the services layer respond to XML commands asynchronously, i.e., once an XML command is received through an HTTP "POST" message, the HTTP communication is first completed before the request is processed. The sender tags each message and upon completion of the command the component returns an XML message along with the identifying tag, indicating either that it has successfully completed or that an error was encountered. We chose this approach as opposed to using a Remote Procedure Call (RPC) approach such as RMI, CORBA or SOAP to facilitate asynchronous communication where the caller does not wait for the completion of the command. Brumitt et al. [15] and Arnstein et al. [16] have elaborated on problems related to RPC style communication in ubiquitous computing environments and have chosen a similar asynchronous approach. Also, using the HTTP protocol allows us to more easily implement a distributed system and configure the machines on which a service runs.

The Services Layer has six types of components that have been implemented to provide an initial basis for applications that use Steerable Interfaces. The Vision, Projection and Steerable Camera interfaces afford steerable visual input and output, while the 3D Environment Modeling, User Localization and Geometric Reasoning components enable tracking of individuals in the space and reasoning about individuals and the spatial relationships they have with interaction surfaces in the environment. These last three components have been included because they are especially useful in achieving adaptation of the interface to user and environmental context. Brumitt et al. [17] have articulated a common view regarding the importance of geometry and geometric reasoning in their EasyLiving [18] project. In our system, the tracking, modeling, and reasoning components provide critical contextual information that allows us to direct the interactive display onto the appropriate surfaces that are most effective for the interaction required. For instance, occlusion, which is often a problem in steerable interfaces, can be detected by geometrically reasoning about the position of the user and the projected surface. If occlusion is determined to occur, the projected interaction can be redirected and the content adapted. We will show how this is particularly useful in the example in Section 6.

The **Projection** component EDML API, called PJML, provides the declarative representation of the extents of display images along with their spatial arrangement in a known frame of reference. The basic PJML action commands enable the instantiation, activation and deactivation of this declarative information. The projection surfaces and associated real world parameters are known only to this component and are also referred to by the application symbolically by name. This leaves the responsibility of mapping the declaratively defined visual widgets to relevant surfaces, as well as the intrinsic implementation, completely to this component. Two basic widgets have currently been defined for this component. An *Image* is basically a bitmap that can be projected onto a surface. A *Stream* is a rectangular region defining a portion of the display buffer on the machine that is running the Projection component. This region can be dynamic visual imagery that is then mapped onto the surface designated via PJML.

The **Steerable Camera** component EDML API, called CAML exposes an interface that allows the application to steer the camera to a named location and adjust all the relevant parameters, such as, zoom, focus and gain to access video data. As in the case of the other modules this is a general-purpose component that can be used by applications to capture and use video data as is required.

The **Vision Interface** component EDML API, called VIML, provides two very important functions. First, it enables the declarative description of the widgets of a specific visual interface interaction along with their spatial arrangement in a known frame of reference. The basic VIML action commands enable the instantiation, activation and deactivation of this declarative description. The interaction surfaces and associated real world parameters are known only to this component and are referred to by the application symbolically, by name. This leaves the responsibility of mapping the declarative interface description to relevant surfaces completely to this component. Second, it provides a mechanism for returning events to applications that relate to the users manipulation of a widget - such as triggering a button press. All the necessary usage and environmental details are hidden from the application developer, who is concerned only with the arrival of an event associated with a specific widget. The component implementation is free to change allowing us to explore alternative approaches for recognizing gestures.

We have currently defined and implemented three vision interface widgets, a *Button*, a *Slider*, and a *Track Area*. Buttons are simple visual objects that are triggered when "touched". Sliders allow a user to adjust a continuous single dimension parameter by moving their finger along a designated rectangular region. Track Areas allow a user to adjust a continuous two dimension parameter (X,Y) by moving a finger within a rectangular region. The Projector component can endow widgets with a visual appearance by projecting imagery that coincides with the position of the widget in the camera's field of view, but this is not a necessity. Widgets can be tangible objects, such as real Buttons, or Sliders, or even pictures of widgets. This component's responsibility is to provide the basic functions for recognizing gestures in a ubiquitous computing environment.

Figure 2 graphically illustrates the mapping process that occurs when an application combines the function of the Projection component, the Vision Interface component and the Steerable Camera component to create an *interactive display* in the real world. The Definition process highlights that a specific interaction has a single declarative display and interaction area. During the Mapping phase the Projection and Steerable Camera components steer their associated devices to the surfaces named in the definition. H and Hv are the homogeneous transformations that define the mapping of the display and interaction areas to projection and interaction surfaces respectively. These transformations for defining surfaces are derived from a calibration that is first performed on the respective host machines upon which the Vision Interface, the Steerable Camera, and Projector components are running.

The **3D Environment Modeling** component is a modeling toolkit for creating and manipulating an environment model to support steerable interfaces. It is in many ways a simplified version of standard 3D modeling software. It supports basic geometric objects built out of planar surfaces and cubes and allows importing more complex models. However, the toolkit provides additional objects such as projectors and projection display surfaces and annotation capabilities that are specifically required for steerable interfaces. Almost every object in the model can be annotated. This makes it possible to attach semantics to objects such as optical properties of a surface and its preferred usage. The toolkit stores the model in XML format, with objects as tags and annotations as attributes. This format allows the model to be easily defined and manipulated by applications in the architecture. The modeling toolkit is also designed to be accessible to the Geometric Reasoning component described below.

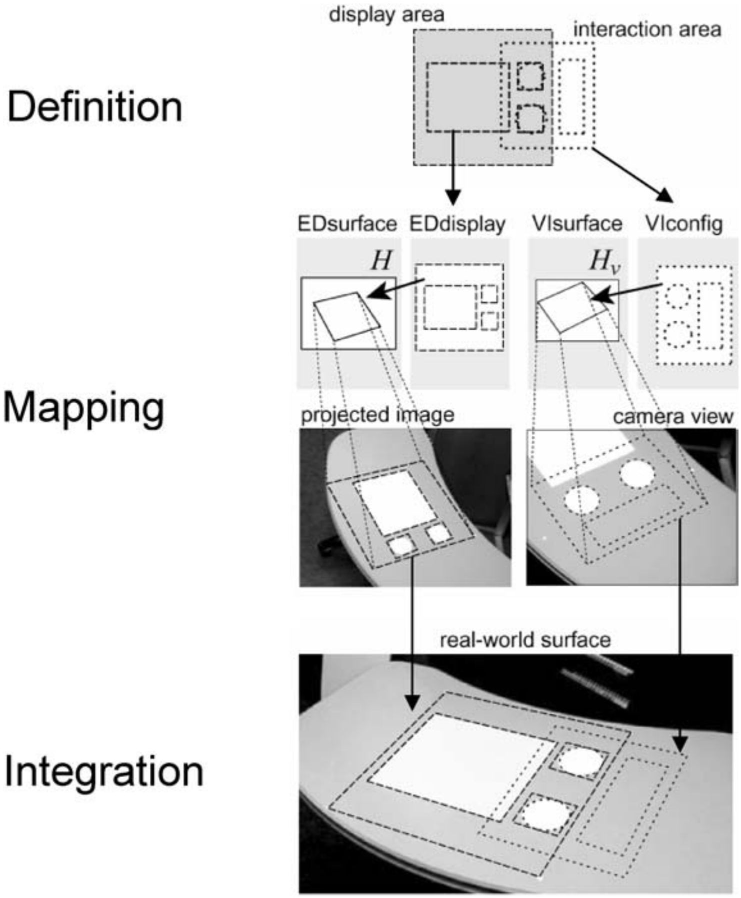


Fig. 2. The process of mapping the declarative display and interaction area definitions onto real world surfaces.

The **User Localization** component keeps track of the 3D position of a user's head in relation to the environment model. In our current implementation, we use multiple cameras with overlapping views to triangulate and estimate this position. Our system tracks user position to an accuracy within a few inches and operates at video frame rate. This allows us to reason quite effectively about the appropriate surface for displaying the interface and about user occlusion of the display, as discussed above. The EDML API for the User Localization component is called TRML, and provides a basic structure for queries about the user's presence, position, and orientation in the environment.

The purpose of the **Geometric Reasoning** component is to facilitate intelligent selection of the appropriate display and interaction surfaces based on criteria such as proximity to the user and non-occlusion of the surface by the user or by other objects. Applications or other modules can query the geometric reasoning engine through the EDML API, called GRML. The API currently supports two types of queries. The first type of query is a property look-up action. The geometric reasoning engine returns all the properties of a specified display surface given the location of the user in the environment. The user location is typically obtained by the application from the User Localization component described above. Other possibilities include simulated user positions for planning and testing purposes. In the second type of query, the reasoning engine receives a user position and a set of criteria, specified as desired ranges of display surface properties, and returns all surfaces that satisfy those criteria. Properties of a display surface include:

1. Physical size of the display surface.
2. Absolute orientation angle between the surface normal and a horizontal plane.
3. Distance between the center of the user's head and the center of a surface.
4. Position of the user relative to the display surface, defined as the two angles to the user's head in a local spherical coordinate system attached to the display surface. This indicates if the user is to the left or to the right of a surface.
5. Position relative to the user, defined as the two angles to the display surface in a local spherical coordinate system attached to the user's head.
6. Occlusion percentage, defined as the percentage of the total area of the surface that is occluded with respect to a specified projector position and orientation.
7. An occlusion mask, which is a bitmap that indicates the parts of a display surface occluded by other objects in the model or by the user.

2.2 The Integration Layer

The Services Layer described above has been designed to meet some of the important requirements listed at the beginning of section 2. Each component in the Services layer has a very specialized function in a particular aspect of a ubiquitous computing environment. Through EDML, the layer provides one level of abstraction between function and implementation of components to facilitate replacing the underlying components that realize a function without affecting applications that use the component.

However, *coordination* of these components to deliver new functionality is beyond the scope of the Services Layer. Hence, we defined an Integration Layer that has a built-in set of classes to facilitate the integration of the underlying components in the Services Layer. We decided that JAVA would be a powerful programming language to use in defining this set of classes for developing ubiquitous applications. Our goals here were to provide a set of basic classes that could be used by an Application

Framework to provide a set of concepts that would facilitate development. The requirements we derived for this layer are:

- Provide a JAVA based API for all EDML objects and associated commands
- Facilitate communication with the services layer by providing a set of specialized servers to handle message input and output as well as events returned
- Handle event propagation back to the appropriate widgets
- Provide simulation capability for each service, enabling development without the need to be connected to the precious resources of the space
- Handle synchronous and asynchronous message passing and callbacks as well as coordination or synchronization of sets of asynchronous calls

Figure 1 illustrates the classes that were built to satisfy these requirements. The **Servers** block consists of a set of classes for generating servers (as well as specific server instances) that handle serialization of messages sent to each component, insuring that the order of arrival is the same as the order of transmission. This is quite important in cases where one command to a component must precede another. In addition, these servers also handle messages returned from components signaling success or failure in execution of the requests and events returned from the vision interface for propagation to the application.

The **PostOffice** class allows the JAVA based EDML commands to either run synchronously, that is, wait for the return result of the requested command, or asynchronously, with the ability to get a callback upon completion of the command. The PostOffice uses a simple mechanism, similar to the concept of "Return Receipt" common in real post offices. The idea is that each request is tagged with a unique ID. If a synchronous call is requested, the calling thread issues a *wait*, and proceeds only after the post office *notifies* it of the receipt of a "return receipt" that has the unique ID. Similarly, asynchronous calls would just proceed, but would have a callback method called upon receipt of the "return receipt" that had the appropriate unique ID. Note that messages could share a common "return receipt" ID, in which case a thread could wait for the completion of a set of asynchronous commands, a very useful and frequently used feature.

The **EDML Classes** allowed a JAVA program to easily construct all the objects defined by EDML and to issue synchronous or asynchronous calls to the components in the Service Layer. Visual interaction widgets are subclasses of JAVA swing classes. For example, buttons are a subclass of *JButton*, so they would inherit all their capabilities, again a very useful feature, which we will discuss in the next section. Similarly, display widgets are inherited from relevant swing classes. For example, an Image is a subclass of *Jlabel.* To invoke the behavior of a Button the Event Manager simply needs to call the *doClick* method of the Button. The advantage of this approach is that the same functionality can be triggered through several means.

For example, consider a media player application that can be moved to different surfaces in space, allowing the user to play, stop, rewind etc., by touching projected buttons. Our approach allows this media player application to be controlled either through gestures recognized through the camera based Vision Interface, or simply through an alternate on-screen GUI.

To facilitate development, we have also incorporated Simulators to the integration layer framework. **Simulators** allow application development to proceed without the need to actually be in the physical space. Each component has a corresponding simulator class that could be used to simulate receipt of messages and returns that would normally be generated by the actual component. Also, an event simulator is available for transmitting simulated events, which is again, very useful in developing and debugging an application. Several levels of debug traces can be switched on and off, providing a complete trace of the system.

The second arrow from the left, labeled JAVA (EDML Adapter API) in Fig. 1 indicates the path that a JAVA application could take by calling the set of JAVA EDML classes. Taking this direct path to the EDML classes, however, can be tedious and requires coordination of numerous calls and a significant amount of code to create simple applications. Note that an application that generates an interactive display would have to individually coordinate all of the components in the layer below. In particular, rapid prototyping can be hard to achieve using the EDML JAVA classes directly. To address this problem, we have developed a framework to simplify the development of Steerable Interface applications, as we will shortly discuss in Section 3.

2.3 The Application Logic Layer

The Integration Layer facilitates communication with the Services layer and coordination of multiple services, as described above. The final layer in our architecture is an Application Logic Layer where an actual application is instantiated, resulting in the instantiation of the corresponding classes in the Integration Layer. The framework for developing such an application is discussed below. This framework was used in a JAVA Application Development Environment that enables graphical composition of the GUI to significantly simplify the whole process of development.

3 Steerable Interface Application Programming Framework

We developed a programming framework consisting of a set of classes that conceptually and practically simplify the task of application development. The current framework consists of over 130 classes including classes in the Integration layer. However, the framework enables applications to be written using as few as 4 or 5 classes. We now outline the major classes that simplify application development.

The primary class for creating an interactive display is called **Interaction**. It contains all the necessary information required by the Vision Interface, the Projector and the Steerable Camera to create a declarative definition of display widgets (e.g. images and streams) as well as vision interface widgets (e.g. buttons, and sliders) on a given surface. In addition, it contains the procedural information defining the behavior triggered by user interaction. It contains at least 20 instances of EDML objects that specify attributes such as the projection and vision surfaces, the container object that hold the display and interaction widgets, along with the widgets themselves, and many other parameters. It basically encapsulates everything needed for one projected interactive dialog. It would take many JAVA calls to instantiate an Interaction, and requires detailed knowledge of layout to provide coordinates for these objects, and to fill out the many parameters. This process is akin to building a traditional GUI, only more complex due to the coordination of a greater number of components. Fortunately, Application Development Environments (ADE) have solved this problem before, through the use of graphical GUI construction tools. Using the Composition Editor (a JAVA bean editor) in our ADE, an Interaction can be completely defined graphically. Stubs for actions to perform are also generated in the process and need to be fleshed out to define the behavior. An application developer would use this tool to generate a number of interactions for a particular application. We will describe the development process in Section 4.

The main driver for an application is the **Application** class. The set of Interactions used by the application are added to an Application instance. The Application instance contains the logic of an application and can maintain the relevant application state required in making any decisions. An Application can define a new **SequenceManager** that is responsible for sequencing the order that Interactions are presented or instantiate one from an existing set of available managers. A simple SequenceManager may call a method in the Application or the current interaction to compute the next Interaction to instantiate.

Once the Application is ready to instantiate an Interaction, it automatically passes it to the **InteractionManager,** which is responsible for orchestrating all the EDML calls that are necessary for bringing the interaction to life. Figure 1 shows this path that completes a description of the three ways that we have to generate Steerable Interfaces. Using this approach requires knowledge of only a few Framework classes and simplifies the task of application development, especially when using the Composition Editor described in Section 4.

A few other classes are useful for tracking a user in the environment and triggering behavior based on position. A **Tracker** class is provided that monitors the position of the user in the environment at a specified frequency. It uses the Observer/Observed pattern to publish location information to all subscribers. A **Zone** class defines a region in space and is an Observer of the Tracker class. It has several methods that fire when a user enters a Zone or leaves a Zone. This class currently has 2 subclasses that implement **Circular** Zones and **Rectangular** Zones. Zones can be added to Applications to automatically trigger behavior appropriate to a location in space.

Finally, a class called **TrackerGUI** allows simulation of the movement of a user in the environment through a mouse-based GUI and a simple graphical representation of the physical space. In this case, the **Tracker** simply gets its tracking information from the TrackerGUI as opposed to the User Localization component. We stress the importance of simulation in frameworks for ubiquitous computing environments. Not only does it allow development to proceed without requiring the resources of the space, but it also helps in debugging the overall complex solution and track down bugs in the underlying components. Much of the Integration Layer and application Development Framework was developed before the Services Layer components were completed and debugged. Also, debugging and integration of the whole architecture was significantly simplified. In Section 5, we will describe how all the classes described above were used to implement an interactive advertising display one might encounter in a retail store.

Fig. 3. The Composition Editor in our JAVA Application Development Environment.

4 Java Application Development Environment (ED-JADE)

Based on the framework described in Section 3, we developed a Java Application Development Environment that enables graphical composition of Steerable Interface applications. Figure 3 shows the Composition Editor, available in our ADE, that we used to create Interactions. The left-hand column of the GUI contains JAVA Swing classes we have sub-classed. The center region is the design window. The dialog on the design window is a visual representation of the interaction we are building. An application developer would populate this dialog with Images and Streams that would be available in the dialog by simply clicking on the appropriate widget and dragging it onto the dialog. Similarly, the developer would select interaction widgets from the palette and place them in the appropriate location. A property editor is available for editing all the relevant fields in the interaction. The property editor, for instance, could be used to define the image that is projected for the Button or the Track Area. In addition, a default display surface could be specified indicating where this dialog should appear in the environment. Using this approach a developer can quickly define the widgets and their spatial relationships without resorting to code.

5 An Example Application

Figure 4 illustrates one example application that was developed using the framework described in the previous sections. This application is a very simple demonstration of the use of a steerable interface in a department store of the future. This example interface combines steerable projection, gesture recognition, user tracking, geometric reasoning, and environment modeling capabilities described in Section 2. In the top left picture an advertisement is presented to the general public as they pass by and tries to lure them toward the display by indicating the word "GIFT" in a large font. As a shopper approaches, the application detects the presence of the shopper and adds some smaller text which is now viewable from this new position saying "A GIFT of beauty" just for you (Fig. 4 – top left). As the shopper approaches even closer a new image is presented showing the potential free gifts available, along with a button that can be pressed to obtain a detailed description of the gifts possible (Fig. 4 – top right). However, in this situation the shopper is in a position that occludes the display. The application reasons about the occlusion and moves the display to a new unoccluded surface near the customer and adapts the interface by positioning the button to a location that is easily accessible (Fig. 4 – bottom left). The shopper, being interested in seeing more details, presses the button and is presented with an additional description of what they can get for free (Fig. 4 – bottom right).

The whole application above was developed with less than 200 lines of code using the application framework and the ADE described in the previous section. In this case, since the interaction could be handled within a single surface (about a 4 foot square projection area) only one Interaction needed to be defined. To implement this application, all we had to do was to instantiate an Application class object and define and add an Interaction containing three rectangular zones. A simple observer class was written and added to the Tracker. Its function was specifically to detect occlusion based on customer position and adapt the image appropriately.

To deploy the application in a space, the Vision Interface, Projection and Steerable Camera components were calibrated to one single surface of our laboratory environment using the calibration GUI tools of the respective components. The position of the surface was also entered in the database of the 3D Environment Modeling component. The application runs continuously, once started, luring those that pass by in our lab to interact with it.

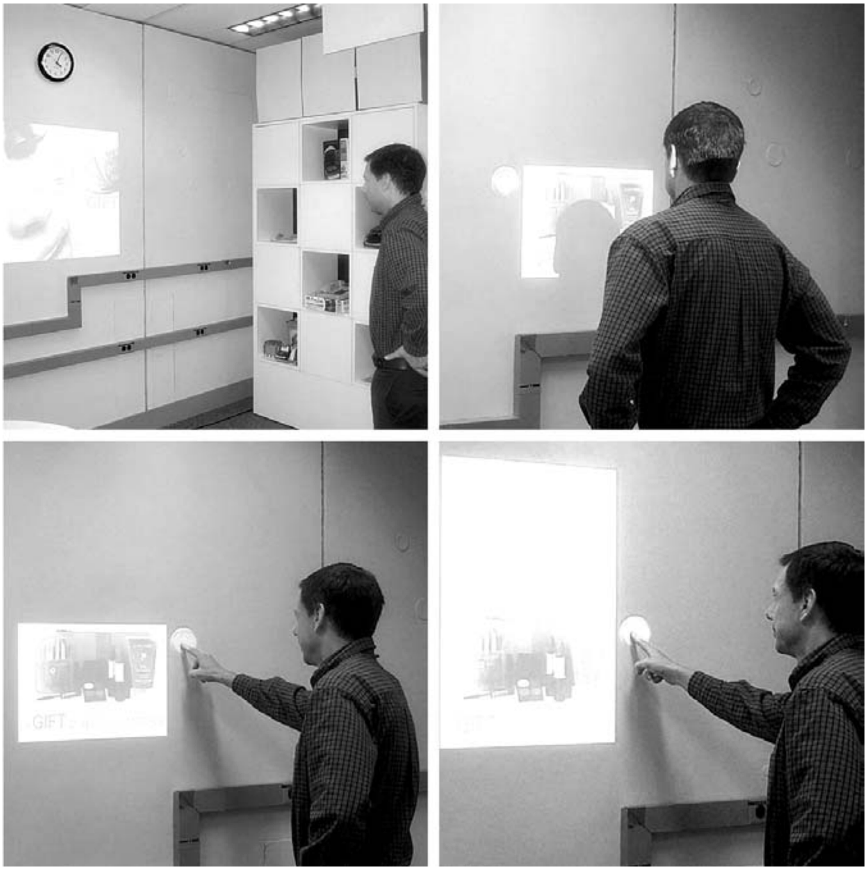


Fig. 4. An interactive display demonstration for a product in a retail store.

6 Discussion and Conclusions

We have presented an architecture and a programming framework for Steerable Interfaces and illustrated their utility through an example application. We believe that our architecture successfully meets the requirements of a Steerable Interface system architecture as outlined in Section 2. The XML approach to the Services Layer results in a distributed architecture that allows the various components to exist on different machines. This approach also facilitates replacement of individual components and exploration of alternate means of achieving the same functionality. The Integration Layer enables coordination, sequencing, and synchronization of the modules provided by the Services Layer, and provides means of simulating these Services. Finally, the Application Programming Framework allows easy development of applications while hiding the complexity of the Services and Integration layers.

Several developers in our group and in other groups have used this architecture and programming framework in different environments. The architecture has enabled eloquent demonstrations of the new styles of interaction offered by the Steerable Interfaces paradigm. However, we recognize that this is still a first attempt at an architecture to support this new paradigm. Below, we summarize some of our observations based on the experience gained thus far in using it.

We have found that the design of the Services Layer and associated communication approach described here has indeed enabled the exploration of several alternative approaches to application development. For example, some of our colleagues have used JavaScript in conjunction with Microsoft's XML/HTTP ActiveX objects to communicate with this layer. Others have used MacroMedia's Flash to link to this layer, while a group at a collaborating university has used JAVA to directly generate XML. Some members of our group rely mostly on developing applications using the JAVA EDML API, while others use the application development Framework extensively.

The extensive use of simulation has been very important in allowing development of the components to proceed in parallel as well as being able to develop applications with out requiring a lot of time in the actual physical space or use of associated resources like the projector and camera. It has been our experience that application development for ubiquitous computing environments in general will have to rely heavily on such simulations.

We realize that adaptability is a critical concern for Steerable Interface systems. The ability to effectively use contextual information to determine the most appropriate surface for interaction as well as adapting the content based on this context is crucial. The support for geometric modeling and reasoning in our architecture has allowed us to develop several applications that exhibit such adaptability. In particular, we are able to address one of the most important issues in Steerable Interfaces, namely occlusion.

We have worked with other colleagues to develop a prototype workspace environment called BlueSpace[14] using the framework described in this paper. This environment has been demonstrated extensively at several research centers around the world. Recently, this system was demonstrated continuously for nine days at CeBIT 2003 in Hanover, Germany and proved to be quite robust. The Steerable Interface technology was very well received by the business community, the media and by every day people that attended – a positive indication of the potential of this paradigm.

In the future, we plan to continue to evolve this architecture to support the growing needs of Steerable Interface applications. At the Services Layer, we plan to incorporate the features of the latest projection devices, as this is an area that is advancing quickly. Adding steerable audio input and output functionality for voice and sound are another area we plan to explore. This will focus our attention on issues related to multi-modality in our architecture. At the Integration Layer, we also plan to extend and test the architecture to address the coordination of numerous steerable input and output units over large spaces. Support for remote monitoring, control, and diagnosis are other challenging architectural issues. Other issues of interest go beyond the architecture to the underlying science and technologies for Steerable Interfaces. Simplifying or minimizing the need for calibration is a very important area to ensure that these kinds of applications abound. Understanding human perception of surfaces and objects that are being projected upon will allow the selection and adaptation of surfaces based on their characteristics as well as the projection characteristics. Finally, looking at novel gestural interaction paradigms that are better matched with ubiquitous computing environments is a topic we will pursue.

References

1. Weiser, M.: Building Invisible Interface. Keynote: *User Interface, Systems, and Technologies (UIST)* conference, November, 1994, slides: http://www.ubiq.com/hypertext/weiser/UbiHome.html
2. Dourish, P. Where the Action Is. MIT Press, Cambridge Massachusetts (2001)
3. Ishii, H., Ulmer, B.: Tangible Bits:Towards Seamless Interfaces between People, Bits and Atoms. In: Proc. of CHI'97. Atlanta, Georgia. 1997. pp. 234-241
4. Rehman, K., Stajano, F., Coulouris, G.: The Interfacing with Invisible Computers. NordiCHI, October 19-23, 2002, Arhus, Denmark.
5. Pingali, G., Pinhanez, C., Levas, A., Kjeldsen, R., Podlaseck, M., Chen, H., Sukaviriya, N.: Steerable Interfaces for Pervasive Computing Spaces. PerCom 2003, Ft. Worth, Texas
6. Pinhanez, C., The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Images. Proc. Of Ubicomp 2001, Atlanta, Georgia, September, 2001.
7. Kjeldsen, R., Pinhanez, C., Pingali, G., Hartman, J., Levas, A., Podlaseck, M., Interacting with Steerable Projected Displays. International Conference on Automatic Face & Gesture Recognition, May, 2002, Washington, D.C.
8. Pinhanez, C., Kjeldsen, R., Levas, A., Pingali, G., Podlaseck, M., Sukaviriya, N.: Using Computer Vision in Steerable Applications. Submitted to ICCV 03
9. Sukaviriya, N., Podlaseck, M., Kjeldsen, R., Levas, A., Pingali, G, Pinhanez, C.: Embedding Interactions in a retail Store Environment: The Design and Lessons Learned. To appear in INTERACT03 Ninth IFIP TC13 International Conference on Human-Computer Interaction, September 2003
10. Sukaviriya, N., Podlaseck, M., Kjeldsen, R., Levas, A., Pingali, G, Pinhanez, C.: Augmenting a Retail Environment Using Steerable Interactive Displays. In Proceedings of CHI 2003 Conference on Human Factors in Computing Systems, April 2003.
11. Kjeldsen, R., Levas, A., Pinhanez, C.: Dynamically Reconfigurable Vision-Based User Interfaces. 3rd Intl. Conf. on Computer Vision Systems, April 2003, Graz, Austria.
12. Silverman, H., Patterson, W., Flanagan, J. and Rabinkin, D. "A Digital Signal Processing System for Source Location and Sound Capture by Large Microphone Arrays", In Proceedings of International Conference on Acoustics Speech and Signal Processing ICASSP 97, Munich, Germany, April 1997.
13. Pompei. F. J.: The Audio Spotlight. Journal of the Audio Engineering Society, Vol. 47, pp. 726-731, October 1999.
14. Lai, J., Levas, A., Chou, P., Pinhanez, C., Viveros, M.: BlueSpace: Personalizing Workspace Through Awareness and Adaptability. In International Journal of Human-Computer Studies, vol. 57(5): 415-428, 2002.
15. Brumitt, B., Meyers, B., Krumm, J., Kern, A., Shafer, S.: EasyLiving: Technologies for Intelligent Environments, In HUC, pages12-29, 2000.
16. Arnstein, L. et al., Systems Support for Ubiquitous Computing: A case Study of Two Implementations of Labscape. Proceedings of the First International Conference on Pervasive Computing, 2002, Springer-Verlag, Germany
17. Brumitt, B., Krumm, J., Meyers, B., Shafer, S.:Ubiquitous Computing & the Role of Geometry. In IEEE Personal Communications, August 2000.
18. Shafer et al. The Easy Living Intelligent Environment System. In Proc. of CHI Workshop on Research Directions in Situated Computing, April 2000.

Evaluation of Visual Notification Cues for Ubiquitous Computing

Peter Tarasewich[1], Christopher S. Campbell[2], Tian Xia[1], and Myra Dideles[1]

[1] College of Computer and Information Science
Northeastern University
360 Huntington Avenue, 161CN, Boston, MA 02115
{tarase,tianxia,cdideles}@ccs.neu.edu
[2] IBM Almaden Research Center
650 Harry Road, NWE/B2-229, San Jose, CA 95120-6099
ccampbel@almaden.ibm.com

Abstract. With increased use of mobile information technology and increased amounts of information comes the need to simplify information presentation. This research considers whether low-information-rate displays (such as those used in mobile devices) can provide effective information awareness. An experiment was performed to measure the performance/size tradeoff of visual displays ranging in size from two LEDs to nine LEDs, and using a number of display characteristics – i.e., color and blinking in various combinations. Results show a reliable tradeoff between performance (participant response time and accuracy) and display size (number of LEDs). However, even the full set of 27 messages can be conveyed with high recognition accuracy using only three LEDs by mapping the messages into color and position. Thus, mobile devices with micro-level form factors can be designed to convey critical information and provide effective notifications. Future work and a prototype developed from this work are discussed.

1 Introduction

With the growing use of mobile information technology comes an increased concern about the impact of such technology on society as a whole. Many computer-based devices are no longer constrained to relatively permanent work or home environments, but are likely to be found in almost any physical location or in any social setting. Concerns with mobile applications have arisen in areas such as safety (e.g., using a handheld organizer while driving) and appropriateness (e.g., using a cell phone in a restaurant). Many organizations have even started regulating the use of mobile devices in specific locations and under certain circumstances, sometimes using technology itself (such as cell phone jammers) to enforce "proper behavior". With an increase in mobile devices also comes an increased risk of "attention overload," which can occur when individuals are overwhelmed and interrupted by intrusive and attention demanding external events, such as loud auditory cues or flashing bright lights [4].

A.K. Dey et al. (Eds.): UbiComp 2003, LNCS 2864, pp. 349–366, 2003.

In such technology-laden environments, information overload is a serious problem as well. While information is necessary to perform many tasks, the human mind is limited in terms of how much information it can process at one time. The problem of information management becomes even more difficult and complex in mobile environments. People must juggle a multitude of dynamic sights, sounds, and other stimuli that convey information and compete for their limited attention. One way to reduce information overload is through the use of meta-information, which can require less effort to process and can result in fewer or less severe disruptions. If meta-information is deemed important, the person receiving it can make a decision whether or not to seek additional details. For example, a mobile worker may not need or want the entire contents of a message or an announcement every time one becomes available. It may be too distracting (and perhaps too dangerous) to the worker's primary tasks. However, they may wish to receive a notification that a message is available, along with an indication of how important it is, and its source. That way, the worker can make their own decision, based on their current situation, whether or not to stop their primary task to access the contents of the message.

This research investigates the design and use of notification cues, which indicate the status or availability of information that is of interest to a particular user in a ubiquitous setting. Notification systems must "present potentially disruptive information in an efficient and effective manner to enable appropriate reaction and comprehension" [16]. Examples of notification cues include the ringing of a cell phone for an incoming call and the chime on a handheld device when a text message arrives. Notification cues are a form of information, and questions arise such as what form these cues should take and how appropriate they are in different settings. Determining notification cues for use in ubiquitous environments can become quite complex, requiring the selection of appropriate delivery channels based on continuously changing contexts and dynamic information needs.

Specifically, this paper presents the results of an experiment that measured the comprehension of and preferences for different visual displays. Each display conveyed the same amount of information, but differed in the number of lights (LEDs) used, their physical arrangement (pattern), the colors used, and whether the lights blinked or not. More lights used in a display means that more room is needed on a device that conveys the notification cue. More lights, however, and the pattern of those lights, may convey information more quickly and easily to the user. Blinking lights and different colors may add complexity to the cue, and may make it more difficult to understand the cue, but allow for a more efficient (i.e., compact) display. This experiment, therefore, investigated whether or not tradeoffs existed between several different cue displays.

The paper is organized as follows. Section 2.0 provides some background on notification cues, including material on human attention and distractions. Section 3.0 describes the methodology used in the experiment, including a description of the configurations and reasons behind the choices made for each display. Section 4.0 presents the results of the experiment, and section 5.0 discusses the results. Section 6.0 draws some conclusions and provides directions for future research.

2 Background

Notification cues can take on various characteristics. Cues can be visual, auditory, tactile, or multimodal in nature. They can be private such that only the receiver is aware of then, or public such that everyone in the immediate vicinity will receive the cue. Cues can also range from being quiet and subtle to being loud and intrusive. A ringing cell phone is an auditory, intrusive, and very public notification cue. A vibrating cell phone is a tactile, subtle, and private notification cue that can convey the same information [4]. There may be different situations where the use of each of these cues is more appropriate. Notification cues also need to safely compete for a user's attention in a world full of an increasing number of distractions, especially in the mobile environment. Important information needs to reach its intended user quickly. Critical notifications that are delayed too long or fail to be delivered can lose their value completely.

The design and use of notification cues must take into account the intricacies of human attention in dynamic environments. Attention involves the allocation of perceptual or cognitive resources to something at the expense of not allocating them to something else [5]. Humans have a limited amount of resources available for allocation to different tasks, therefore everything cannot be attended to at once. People can attend to a modality (vision, hearing, touch, taste, smell), a color, a shape, or a location [5]. The decision to attend specifically to one of these over the others arises from the task at hand. However, events occurring in the unattended modalities will not go unnoticed [15, 18]. For example, when reading a book, a person may ignore most sounds but will respond if their name is called. The brain processes unattended modalities at a level that allows for recognition but keeps them unnoticed for the most part [23].

With computer applications that are used in the office, home, or similar settings, the context is known and is relatively stable from minute to minute. While this does not mean that there cannot be multiple activities competing for a user's attention (e.g., animated ads, intrusive pop-under windows, stock alerts, and email notifications), the environment outside of the computer is fairly consistent for a given user from day to day. Most offices and homes function with a fair amount of regularity and predictability, even if they do experience a great amount of activity. The user can devote a relatively consistent amount of attention to actually performing tasks on the computer.

On the other hand, with mobile applications, there can be a significant number of additional people, objects, and activities vying for a user's attention aside from the application or computer itself [21]. Furthermore, since devices are completely mobile, this outside environment can change rapidly from moment to moment. A mobile application may not be the focal point of the user's current activities [7], as the user may be trying to juggle interaction with a mobile device along with other elements in the environment (e.g., walking on the sidewalk of a busy city street with small children while receiving directions from a navigation system). Mobile activities can be complex because of changing interactions between the user and the environment. The amount of attention that a user can give to a mobile application will vary over time, and a user's priorities can also change unpredictably [11].

An environment that consists of too many distractions can be confusing and unmanageable. Hansson, Ljungstrand, and Redstöm [4] defined *attention overload* as a situation where "people are overwhelmed and interrupted by obtrusive and attention demanding external events, such as loud auditory cues or flashing bright lights." Notification cues have to be designed and used such that they minimize the possibility of overloading the attention of the intended recipient and any surrounding people. Otherwise, the cues may prove to be ineffective or may be ignored completely.

Much research effort has been devoted recently to studying notification systems in the form of secondary displays, or peripheral displays, which provide information to the user that is not central or critical to their current or primary task. For example, a user may have a one line display on their computer screen in which scroll current news headlines. Studies have looked at the effectiveness of presenting information in secondary displays in various formats [1, 10, 14, 16, 24]. The Scope notification summarizer [24] used color and relative location to convey the source, priority, and type of information available to the user. Research has generally found that user performance on primary tasks is negatively impacted by these secondary tasks [1, 14], with some exceptions [16].

Other research has investigated notification systems and devices specifically for mobile environments. Wisneski [27] described a subtle and private notification device in the form of a watch that changes temperature as stock prices change. Holmquist, Falk, and Wigström [8] tested a device called the "hummingbird" that notified its user of the close proximity of other group members by producing a sound ("humming") and listing the identities of the group members.

In a first attempt at creating a subtle notification cue that was also public, Hansson and Ljungstrand [3] created a "reminder bracelet", worn on the user's wrist, which notifies a user of upcoming events (e.g., meetings). The bracelet consists of three light emitting diodes (LEDs) that are triggered progressively as an event draws closer. The notification information comes from a PDA (carried by the user) that is wired to the bracelet. With this device, other people interacting with the user can clearly see that the user is being notified about something.

The delivery of notifications can sometimes be improved by interpreting the current context of the user through the use of sensors and by determining the priority of the information being sent. Nomadic Radio [19] is an auditory notification device that manages voice and text messages in a mobile environment. The form of notification is chosen based on the content of the message, whether or not the user is speaking, and the user's responses to previous messages. Horvitz, Jacobs, and Hovel [9] described the concept of attention-sensitive alerting, whereby the costs of a user's interruption are balanced with the costs of deferring alerts. Horvitz, Jacobs, and Hovel looked at probabilistic models that can make inferences about a user's attention under uncertainty. In addition to this, they created a system to automatically classify the criticality of the alert (in this case, email messages).

Schmidt et al. [20] have experimented with a cell phone that changes its ring type (vibrate, quiet ring, loud ring, silent, or mixed-mode) based on the context of the phone. A sensor board was created to measure light level, tilt, vibration, proximity, temperature, pressure, and sound level. The readings from the board were interpreted by software to provide information current situation of the phone, and its ring was adjusted accordingly. For example, if the phone was judged to be on a table, it would ring quietly (the assumption being that the user is in a meeting).

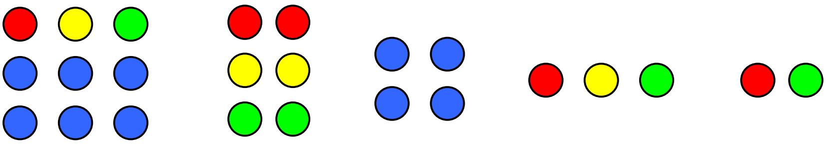


Fig. 1. The set of cue displays (low-information-rate displays) used in the experiment. Display 5 uses multi-color LEDs (red/yellow/green)

3 Evaluation of Low-Information-Rate Displays

One key issue in designing mobile notification displays is how to adequately inform users given a small or even micro form factor without requiring a great deal of attention or needing long training sessions to learn coded messages. While very small LCD screens have been developed (e.g., watch displays), lower information rate displays such as LEDs have the benefit of (a) requiring less cognitive effort to understand (i.e., less distraction), (b) allowing for smaller and even micro level form factors, and (c) using less power. Simply speaking, the less information conveyed, the less attention required to use that information. However, less information does not mean the message is not informative. Even small amounts of critical information can be highly informative and keep the user aware in mobile situations.

An experiment was conducted to test comprehension and preferences of notification cues across five different cue displays (i.e., low-information-rate displays) ranging in number of LEDs from two to nine (see Figure 1). The number of LEDs used is related to the size of the display and the complexity of the display is related to how many different states each LED can assume. As complexity increases, performance should decrease. Because the same number of messages must be conveyed by all the cue displays, complexity will increase as size decreases. This is a critical tradeoff for mobile displays that attempt to minimize size. The goal of the present experiment was to determine the function of this tradeoff and find the optimal point on this function. This optimal point will show the smallest array of LEDs that can be used while still maintaining high performance and maximizing preference. In other words, we assume that as size decreases, complexity increases, and performance will decrease. If the relationship between size and performance is linear, then no optimal point can be found. A non-linear relationship with a knee in the curve, however, would indicate there is an optimal design point that could be taken into account during system design.

Identical messages were mapped into each cue display and consisted of three cue dimensions at three levels each for 27 distinct messages (see Table 3). The messages were mapped into the cue displays using position, color, and blinking.

Subjects were given a pre-experiment questionnaire concerning current mobile technology use and demographic information. Subjects were then presented with each cue display showing one of the twenty-seven messages and asked to indicate the priority, source, and descriptor of the message. Response accuracy, start time, and end time were recorded. Finally, subjects were given a post-experiment questionnaire to indicate which cue display they preferred.

3.1 Complexity and Mapping Functions

Assume a *cue display 0* consisting of five LEDs in a single row. Each LED has ON and OFF states, corresponding to the numbers “1” and “0”. If we think of binary numbers represented by these five LEDs, the following mapping function is possible:

Table 1. Example of a complex mapping as demonstrated by a binary coding of messages across five LEDs in a row

Binary code	Cue value (Priority level, Source, Descriptor)
00001	High, family, reminder
00010	High, family, news
00011	High, family, email
00100	High, friends, reminder
…	…
11011	Low, work, email
00000	Default state: no information is conveyed
11100 – 11111	(Not used)

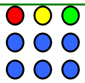
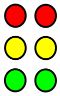
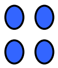
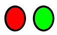
This kind of mapping for conveying meta-information is inappropriate, not only because it is difficult for most people to remember all 27 binary numbers, but also because the mapping between binary numbers and cue values is not intuitive. Interpreting cue display 0 requires scanning through all the binary codes in long-term memory until the correct one is found. This requires scanning one to 27 codes. Before scanning the codes, they need to be recalled from long-term memory and loaded into working memory. Unfortunately, the capacity of working memory is fixed, about five to nine independent items, according to Miller [17]. This means that we could reduce the interpretation time by reducing the information we need to access. This can be achieved through the grouping of related information in cue display designs.

In contrast to the design of cue display 0, which mixes all three categories together and generates 27 instances, we will keep these three categories separate, and interpret the display using three mapping functions corresponding to these three categories. By grouping the information by category, we will have at most nine instances from three functions. We interpret one value for each of three categories, and get a whole cue value simply by combining three values together. Good mapping functions should lead to good display interpretation times.

3.2 Displays Used for the Experiment

Each display conveys the same amount of information, but differs in terms of the number of lights (LEDs) used, their colors, whether they blink or not, the configuration pattern of the lights, and how the information presented on the display is mapped to these characteristics (see Table 2). Using more lights means that more space is taken up by the display that conveys the notification cue. More lights, however, and the layout of those lights, may convey information more quickly and easily to the user. Blinking lights and different colors may add complexity to the cue, and may make it more difficult to understand the cue. This experiment investigates whether or not there are tradeoffs between a variety of visual displays used to present notification cues.

Table 2. Visual qualities and mapping complexity of the five cue displays

Display	No. LEDs	"On"-state Used	"Off"-state Used	Blinking Used	Multi-color LEDs	Mapping Complexity
	9	X				Very Low
	6	X	X			Low
	4	X	X	X		Medium
	3	X	X	X		High
	2	X	X	X	X	Very high

Cue Display 1. The LEDs in the first row are red, yellow, and green; the other LEDs are blue. Values for priority are mapped to the LEDs of the first row, values for source to the second row, and descriptor to the third row. Each position in a row corresponds to a value for that information category. In this example, high is mapped to the first light of the first row, medium to the second, and low to the third. Mappings for source and descriptor are similar. Exactly one LED is lit in each row for any notification cue.

This cue display is the simplest mapping from message to cues. Users do not need to combine several dimensions before comprehending its meaning. The independent items in working memory could be as few as three, because three physically separate elements in the same category that are stored together as a group in long-term memory may be recalled and maintained in working memory as a single entity [26]. It is possible not to store a whole category as one item in memory in this design, because users may not need to process information in each group.

Cue Display 2. LEDs in the first row are red, in the second row are yellow, and in the third are green. The LEDs could also all be the same color. Priority is mapped to the first row, source to the second, and descriptor to the third, as above. This time, however, the row values are indicated by a binary code. Both LEDs lit indicate the first value (e.g., high), just the first LED lit indicates the second, and just the second LED lit indicates the third value.

Cue Display 3. All four LEDs are the same color. The first row is priority, and its value is conveyed through a binary code as in display 2 (11 = high, 10 = medium, and 01 = low). The bottom left LED is source, and the bottom right is descriptor. The values are indicated by blinking, lit, and off (e.g., blinking = family, lit = friends, and off = work). This is different from displays 1 and 2 in that "off" is now used to indicate a value. We could also place all four LEDs in a row.

Cue displays 2 and 3 map the values of each category to the LEDs using more complex encoding functions than the first design. These functions are not straightforward, and the mapping of each basic element needs to be calculated before it is comprehended. So we may not rehearse and maintain 3 basic elements in one category as a single entity. These functions need more working memory, and are assumed to produce a slower response due to the tradeoff between performance and device space consumed.

Cue Display 4. The first position is priority, the second is source, and the third is descriptor. Blinking is the first level of information, lit is the second, and off is the third. While this mapping is fairly intuitive, it requires subjects to associate blinking and on/off together as a single dimension from high to low. Additionally, the blinking may be missed during a hasty response leading to more errors and worse performance.

Cue Display 5. Using color LEDs (e.g., red, yellow, and green in the same LED) one can reduce the number of LEDs down to two. The first LED is message source, with red indicating family, yellow friends, and green work. The second LED is descriptor with the same mapping. Priority is indicated by the blinking of the LEDs (regardless of color). Both LEDs blinking signify high priority, one blinking is medium priority, and none is low. This design does not use grouping of information. We could not use 2 LEDs to represent 3 groups by position alone. Users need to remember all 27 instances of a function. The design will utilize both changing of colors and different blinking rates in order to reach enough complexity to accommodate all combinations.

An attempt was made to use display dimensions in a way that best produces an intuitive mapping. For example, color is an especially effective way of coding nominal sets [13]. Kahneman and Henik [12] and Ware and Beatty [25] pointed out that the human visual system is very effective at distinguishing a small number of distinct colors. Some research [6] points out that red, yellow, and green should be reserved for "Danger", "Caution", and "Safe", respectively. In keeping with this recommendation, red is used for high levels of each cue category (e.g., high priority), yellow for medium, and green for low levels. Finally, only one level of blinking was used as the use of too much blinking, particularly more than two levels, is not recommended [2].

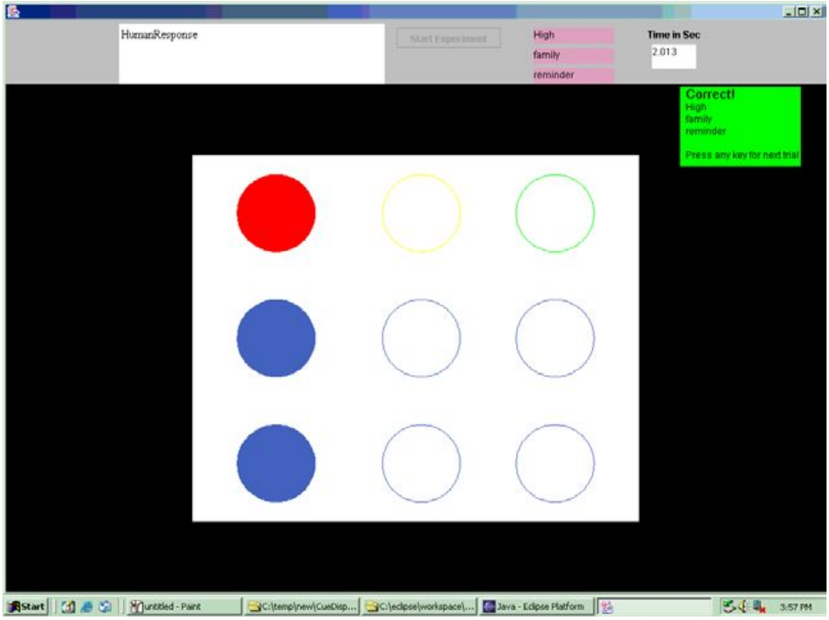


Fig. 2. Screenshot of experiment display, with the stimuli (i.e., cue display) in the center of the screen. The three pink bars at the top of the screen indicate that responses have been made for all three cue categories. Feedback is shown in the green box in the upper-right hand corner indicating that all three responses were correct. Finally, the response time is shown in the white box in the upper right-hand corner

3.3 Method

Subjects. Nineteen college students and one instructor of an information science class at a large university participated in this study. Eighteen were male and two were female. The median age of the participants was 21 years, and each claimed to use a personal computer daily. None of the subjects were colorblind. Nineteen of the twenty subjects carried a cell phone, for 3.2 years on the average.

Design. This was a four-way fully-factorial within-subjects design in which the cue display order was randomized and the trials for each cue display were blocked. In other words, subjects were presented with all the trials of each cue display at the same time (i.e., blocked), but cue display order was random. Cue values were presented randomly without replacement within each block. The four factors included cue display (5 levels), priority (3 levels), source (3 levels), and descriptor (3 levels) with two repetitions for a total of 270 trials per subject.

Equipment and Materials. Subjects completed a pre-experiment questionnaire – a set of questions pertaining to their computer and mobile device background. The experiment was then conducted on a Pentium 4-based personal computer (PC) running Windows XP with the screen resolution set at 1024 x 768 on a CRT screen. A program was written in Java to present the cue displays as animated Graphics Interchange Format (GIF) files. The GIF files were presented on a black background that covered the entire desktop except for the taskbar at the bottom (see Figure 2). The GIF files were 555 X 457 pixels in size, 96 pixels/inch resolution, and had 8-bit color depth. At the top of the screen was a status bar showing what cue responses had been made and the elapsed time for that trial. Subjects completed a post-experiment questionnaire to determine cue display preferences.

Procedure. A training session was first performed where mappings for each of the different displays to be used were shown and explained to each subject. Next, five rounds of testing were performed. In each round, subjects were first shown the mapping for one of the cue displays (randomly chosen) and their responses for the three cue dimensions were recorded. A notification cue conveys meta-information (information about information). For this experiment, the information conveyed by the cues was that summarized in Table 3.

Table 3. Cue categories and associated values. Combining all the values in all possible ways results in 27 different messages

Category	Possible Values
Priority level	high, medium, low
Source	family, friends, work
Descriptor	reminder, news, email

Using all three information categories, there are 27 (3x3x3) possible notification cue values that can be conveyed on any display. These were shown two times each, so a particular display showed a random sequence of 54 notification cue values. These cue values were graphical representations of actual notification cues. They consisted of simulated LEDs, which were simply circles drawn in a particular pattern. “Lit” LEDs were simply fully colored circles. “Blinking” LEDs were circles that alternated between empty and filled. Each display value was shown until the subject entered a response signifying the information represented by the notification cue display, but for not more than 6 seconds. Subjects responded by pressing three keys on the numeric keypad of the PC. Each row of the keypad corresponded to a different information category; row one was priority level, row two source, and row three descriptor. Each of the three keys in each row corresponded to a possible value for that category (e.g., 7 = high, 8 = medium, 9 = low). The first key pressed in each row was taken as the response to that display and could not be changed. At the end of each response, the correct answer was shown to the subject. After the first display, the experiment continued with the remaining four displays (for a total of 270 notification cue values tested per subject). The Java program automatically recorded the subjects’ answers and response times.

4 Results

4.1 Performance

Four measures of performance were collected during the experiment – start-time, completion-time (end-time), difference-time, and percent correct. The start-time was the time that the subject entered the first response on the keypad. The completion-time was the time that the last (third) response was entered. The difference-time was the difference between start-time and end-time. Percent correct was the number of cue dimensions correctly specified. Because there were three responses for each trial, the percent correct was measured as a score between 0 and 3 with 0 meaning no correct responses and 3 meaning all correct responses. A separate four-way repeated measures analysis of variance (ANOVA) was applied to each of these measures. Our alpha rate for all ANOVAs was .01 or 1 percent meaning that the p-value for each test must be less than .01 to be considered significant or reliable.

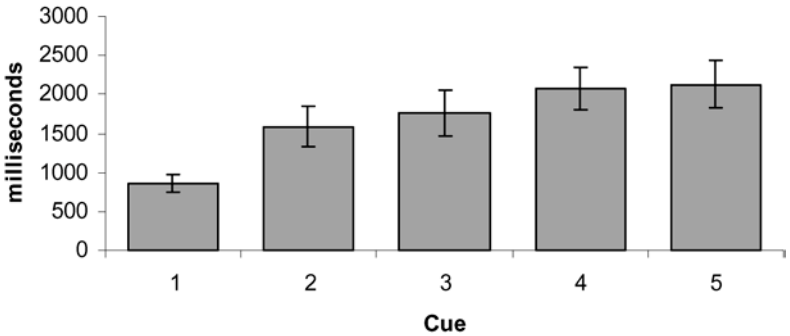


Fig. 3. Start-time increases as mapping complexity and number of LEDs increases

The ANOVA for start-time showed a significant effect of cue display ($\underline{F}(4,72) = 90.23$, $p < .01$) and priority ($\underline{F}(2,36) = 15.66$, $p < .01$). As shown in Fig 3, responses started much faster for cue display 1 (M = 851ms) than the other cues with cue display 5 the slowest. For priority, high (M = 1678ms) and low (M = 1601ms) priorities were reliably faster than medium (M = 1762ms). The error bars in Figures 3 through 6 represent 95% confidence intervals.

The ANOVA for end-time also showed a reliable effect of cue display ($F(4,72) = 72.14$, $p < .01$) and priority ($F(2,36) = 17.68$, $p < .01$) as well as descriptor ($F(2,36) = 7.44$, $p < .01$). Fig 4 shows that cue display 3 has the longest total response time (M = 3645ms) followed by cue display 5 (M = 3568ms) and cue display 4 (M = 3337ms). Similar to start-time, priority for end-time is fastest for high (M = 2982ms) and low (M = 2846ms) than medium (M = 3062ms) priority. Finally, email descriptors (M = 2901ms) were entered significantly faster than reminder (M = 3001ms) or news (M = 2988) descriptors.

The ANOVA for difference-time showed a reliable effect of cue display ($F(4,72) = 34.68$, $p < .01$). In other words, the amount of time it took to enter all three responses varied across cue displays. It took the longest time to enter responses for cue display 3 (M = 1872ms) and the least time to enter responses for cue display 4 (M = 682ms). Also, there was a significant effect of priority ($F(2,36) = 6.74$, $p < .01$) and descriptor ($F(2,36) = 7.24$, $p < .01$).

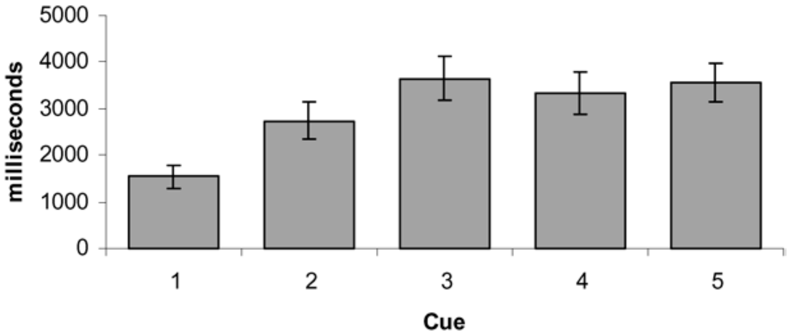


Fig. 4. Completion time is fastest for cue display 1 and slowest for cue display 3

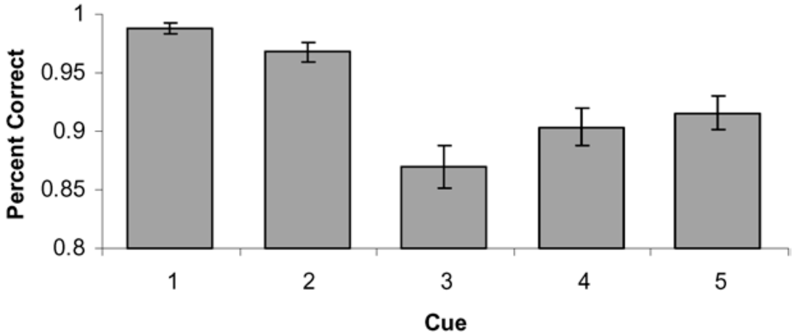


Fig. 5. Percent correct responding is very high for all cue displays but significantly worse for cue display 3

The ANOVA for percent correct showed that only cue display had a reliable effect ($F(4,72) = 6.36$, $p < .01$). Fig 5 shows that cue display 3 had the lowest accuracy while cue displays 1 and 2 had the highest. Overall, the accuracy rates are very high indicating that in the speed-accuracy tradeoff, subjects were focused on high accuracy.

4.2 Questionnaires

Seventeen people ranked the five displays after completing the experimental sessions. In ordering of decreasing preference, subjects preferred displays 1, 2, 4, 3 and 5. Subjects were also asked several more open-ended questions concerning the displays that they had just interacted with. One question asked, if subjects had adequate time to learn any of the displays before using them in the real world, which one(s) would they rather use on a mobile device such as a PDA. Most subjects answered that they would still rather use display one (the nine LED display), with some commenting that the mapping was the most natural and that it was the most easy to use. However, about a third of the subjects commented that displays two, three, or four would be preferable if adequate learning time was available.

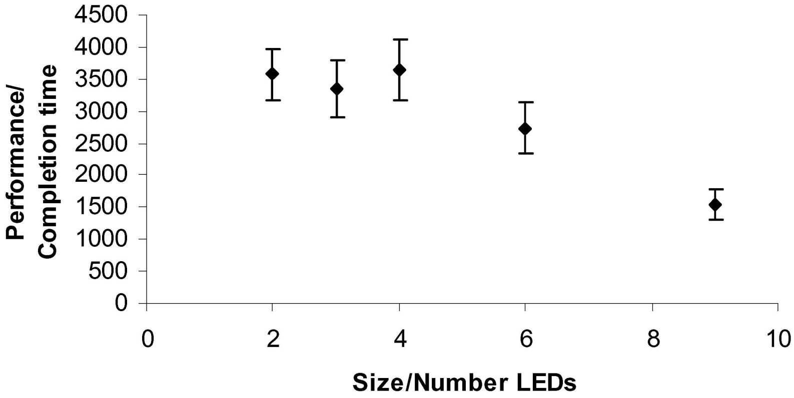


Fig. 6. The performance versus size tradeoff function appears to increase linearly from nine to four LEDs and then plateau

The second question asked that if there was adequate time to learn any of the displays before using them in the real world, which one(s) would rather be used on device such as a watch? For this question, there was a movement in preferences towards smaller displays, with nine people preferring display 4 (three LEDs), three preferring display 5 (two LEDs), and one preferring display 3 (four LEDs). Two still preferred display 1 (nine LEDs), and two display 2 (six LEDs),

The final question asked which displays subjects would prefer not to use again. About half of the subjects commented that they did not like the two LED display (display 5), and a third did not like to use the four LED display (display 3). One person commented that the blinking displays in general were difficult to use.

5 Discussion

As expected, there are explicit tradeoffs between the different displays studied in this experiment in terms of the performance measures collected. What is really surprising, however, is that instead of continuously worse performance, response times flatten out at four LEDs and remain about the same at two LEDs (see Figure 6). In other words, subjects were able to use two LEDs to recognize 27 messages with 92% accuracy, responding in less than 4 seconds – all with little training and only an hour of experience. Part of the explanation for this result is the young subject population used. Certainly, with a wider range of subjects, learning rate could be more of a barrier to using low-information-rate displays.

With cue display 1, subjects performed the best in terms of start time, completion time, and accuracy. This is not surprising, since cue display one was designed to map perfectly with the three message categories and three values for each category. Subjects were able to recognize the cues quickly and accurately. Subjects also seemed to prefer display one overall to the other four displays. Here, the perception may be that if screen real estate is readily available, it should be used for a display that quickly and easily conveys information to the user.

There is a significant increase in both start time and completion time, and a significant decrease in accuracy, for display 2. This display has six LEDs, and while the mapping of its rows corresponds to the information categories, a binary coding system was used to display the three values for each category. Therefore, this increase is not surprising, given the expected extra cognitive processing time that such decoding would incur. Such a display, however, would take up less real estate on a display. A learning curve seemed to factor into the performance with this display, as several subjects mentioned that they would use this one (over display 1) if they had sufficient time to get used to it.

Display 3 was most difficult display for subjects to interact with. Start times were third highest, but completion times were the longest out of any of the displays. Accuracy rates were also the lowest for all of the displays. This display had four LEDs. The top two LEDs were used to represent priority through a binary code, while the bottom two each represented one of the other two categories, with levels represented by on, off, or blinking. There are two possible contributing factors to the unexpectedly high complexity of this display, 1) it combined two different visual representations in one display (binary, blinking) and 2) it used blinking, which seems to be more difficult and time consuming to comprehend by nature. Surprisingly, while many of the subjects did not like display 3, it was not the least preferred display.

While display 4 showed a fairly high start time, its completion time, although higher, was not significantly different than that of display 2. Display 4 consisted of three LEDs, each of which mapped to a category. Levels were indicated by on, off, or blinking. Accuracy rates were not as good as with display 2, but were significantly better than display 3. When subjects were asked which display they would like to use with a watch-sized device (given adequate learning time) most preferred this display.

Display 5 was the display that half of the subjects said they would not want to see again. This two LED display combined the three categories using three colors and binary blinking. The cognitive effort required to process the cues on this display is reflected in the highest start times, and the fourth highest completion times. Accuracy, however, was not significantly different from display 4.

Some of the preferences or performance results found during this study may be related to the characteristics of the display implementations. Priority was always the first row or LED on the displays, so that may explain why the performance was better for priority (or why start time was significantly lower). Priority was also often indicated with color-coding.

Multiple colors are an integral part of all displays except display 3 (the four LED display). The colors may have helped to differentiate the different category values from each other. In the case of display 5, color recognition was necessary to discriminate the different level of two of the categories. When color was used to encode different levels of a particular category, the colors red, yellow, and green were used, similar to a traffic light pattern. These were matched up to priority levels of high, medium, and low, or other category levels matching those going from left to right on the keyboard.

Binary coding was used in displays 2 and 3, where two LEDs were used to convey information about a category level. This may have required additional cognitive effort due to a decoding process needed to match the code to the appropriate level, although an attempt was made to match the codes from left to right with the levels as assigned on the keyboard.

Blinking was also used to indicate category levels in several of the smaller display designs. In displays 3 and 4, LED states of blinking, steady, and off were used to indicate one of the three levels for a particular category. In display 5, blinking of no, one, or both LEDs was used to indicate priority. It may be that for these displays that additional time was needed to recognize whether or not the displays were indeed blinking or not before the entire display was comprehended. Possibly speeding up the blinking rate would allow for faster recognition, but may also incur a distraction and/or annoyance cost.

Finally, it was surprising that significant differences were found for priority and descriptor across reaction time measures and accuracy. The effect was also independent of display. Thus, it may be that the message itself caused the effect. In other words, results showed that subjects were slowest at responding to medium priority messages indicating that medium priority is not very "interesting" or informative. Subjects may prefer only two levels of priority – low and high. The same could be true for descriptor in which the email descriptor was responded to faster than reminder or news.

6 Conclusions and Future Work

In summary, there are several conclusions that can be made from this study:

1. **There is a tradeoff between performance, preference, and display size.** As the number of LEDs decreased, accuracy, speed, and favorable opinion generally decreased. But, performance for the message set used here seemed to plateau at a size limit of four LEDs.
2. **This tradeoff can be mitigated by effective display design.** Using color, blinking, or binary codes appropriately allows the use of fewer LEDs to convey a fixed amount of information.
3. **Position provides the most intuitive representation.** Although position seems to aid message recognition the best, it also has the highest cost in terms of display size.
4. **Multi-color LEDs are useful for conveying more information in a smaller space.** Color was useful for distinguishing between three different types of messages and helps to reduce the number of LED's needed in a display but may not be helpful in conveying a large number of categories.

5. **Blinking may be better for attention getting than providing information.** Performance was not good with the blinking displays and other work [e.g., 2] has shown that few levels of blinking are useful, so blinking may not be a good choice for important meta-information.
6. **The mixing of certain visual factors may degrade performance.** There may be compounding or other effects caused by the mixing of certain visual representations. This is seen by comparing display 3, which had four LEDs but used both a binary representation and blinking, to display 4, which had only three LEDs but used only blinking. Blinking seemed to be harder to comprehend than binary coding, but having both at the same time made cue comprehension even more difficult.

It must be remembered, however, that these conclusions are being made based on the results of this limited, initial study, which is a first step in what is hoped to be a series of related work and additional experiments. This study has looked at a relatively small set of the total number of visual displays available, and there are many alternate implementations of each display. For example, information can be presented using different physical arrangements of LEDs and different binary codes. Learning curves were not investigated as part of this study, and no attempt was made to look at how the results might change over time. The LED displays were also simulated in a laboratory environment, rather than tested in real-world settings using real lights.

Overall, though, it appears that low-information-rate displays can be useful for providing information awareness in mobile situations. With the right display design, the performance reducing and experience spoiling effects of complex mappings can be mitigated. At the same time, this work supports the idea that even the smallest form factors such as watches, rings, and other jewelry can be used as valuable notification systems. This work opens the door for the design of a wide range of low-information-rate notification systems using a variety of information channels including other visual cues (e.g., motion and brightness), tactile information (e.g., vibration and pressure), and auditory information to name a few. Some issues that will be addressed in future work will be (a) measuring the distraction (individual and public) cost of low-information-rate displays in mobile situations, (b) determining what types of messages provide "critical bits" given a certain context or situation, (c) finding the number of messages or level of complexity at which performance breaks down to determine the "upper-bound" in information space for low-information-rate displays.

Other open research issues regarding notification cues include the analysis of end-user requirements and the need to determine exactly what types of information or events that users want to be notified of. The framework for mobile information needs presented in [22] begins to looks at this issue, but much more work is needed. There is also the issue of cue personalization. Allowing someone to customize the form that their notification cue takes may help alleviate some concerns over privacy and security of information, especially if the cue is sent in a public space. A final issue is that the acceptability of a notification cue or device, as with any sort of technology, may depend on fashion or social perceptions. Research in the area of wearable computing will undoubtedly influence the design of notification devices as well.

The results of this experiment are being integrated into the development of a notification cue prototype that alerts the user to approaching events. The events are recorded on the user's PDA, along with a priority for each event. The PDA vibrates when the event comes within a certain amount of time (e.g., half-an-hour) of the current time. A signal is also sent wirelessly to a set of LEDs that form a notification cue conveying a code for the priority of the event and how much time is left until the event. The notification is triggered based on the priority of the event. The PDA can be kept somewhere on the user's body (e.g., in a pocket), while the LEDs can be worn or kept elsewhere. The prototype is therefore provides a multi-modal notification cue (tactile and visual) that is subtle yet public. This work is an extension of [3].

References

1. Czerwinski, M., Cutrell, E. and Horvitz, E. (2000). Instant Messaging: Effects of Relevance and Timing. In S. Turner, P. Turner (eds), People and Computers XIV: Proceedings of HCI 2000, Vol. 2, British Computer Society, p. 71-76.
2. Grether, W. F., and Baker, C. A. (1972). Visual Presentation of Information. In H.A. Van-Cott and R.G. Kinkade (eds.), Human Engineering Guide to Equipment Design. Washington, DC. U.S. Government Printing Office.
3. Hansson, R. and Ljungstrand, P. (2000). The Reminder Bracelet: Subtle Notification Cues for Mobile Devices. In Extended Abstracts of CHI 2000, 323-324.
4. Hansson, R., Ljungstrand, P., and Redstöm, J. (2001). Subtle and Public Notification Cues for Mobile Devices. In Proceedings of UbiComp 2001.
5. Harris, L. R. and Jenkin, M. (2001). Vision and Attention. In M. Jenkin and L. R. Harris (eds.), Vision and Attention, Springer-Verlag, New York, 1-17.
6. Helander, M. G., (1987). Equipment and Workplace Design. In Salvendy, G. (ed.), Handbook of Human Factors, 1st Edition, (pp. 505-620). New York: Wiley-Interscience.
7. Holland, S. and Morse, D. R. (2001). Audio GPS: Spatial Audio in a Minimal Attention Interface. In M. D. Dunlop and S. A. Brewster (eds.), Proceedings of Mobile HCI 2001: Third International Workshop on Human-Computer Interaction with Mobile Devices.
8. Holmquist, L. E., Falk, J. and Wigström, J. (1999). Supporting Group Collaboration with Inter-Personal Awareness Devices. Personal Technologies, 3(1-2), 13-21.
9. Horvitz, E., Jacobs, A., and Hovel, D. (1999). Attention-Sensitive Alerting. In Proceedings of UAI '99, (Stockholm, July 1999) Morgan Kaufmann, 305-313.
10. Huang, E. M., Tullio, J., Costa, T. J., and McCarthy, J. F. (2002). Promoting Awareness of Work Activities Through Peripheral Displays. In Proceedings of CHI 2002, 648-649.
11. Johnson, P. (1998). Usability and Mobility; Interactions on the Move. In C. Johnson (ed.), Proceedings of the First Workshop on Human Computer Interaction for Mobile Devices. Scotland: University of Glasgow.
12. Kahneman, D., and Henik, A. (1981). Perceptual Organization and Attention. In M. Kubovy and J. R. Pomerantz (eds.) Perceptual Organization, Lawrence Erlbaum, Hillsdale, N.J., 1981, 181-211.
13. Mackinlay, J. (1986). Automating the Design of Graphical Presentations of Relational Information. ACM Transactions on Graphics, 5(2), April 1986, 110-140.
14. Maglio, P. P. and Campbell, C. S. (2000). Tradeoffs in Displaying Peripheral Information. In Proceedings of CHI 2000, 241-248.

15. McCormick, P. A. (1997). Orienting Without Awareness. Journal of Experimental Psychology. Human Perception and Performance, 23, 168-180.
16. McCrickard, D. S., Catrambone, R., Chewar, C. M., and Stasko, J. T. (2002). Establishing Tradeoffs that Leverage Attention for Utility: Empirically Evaluating Information Display in Notification Systems. International Journal of Human-Computer Studies, in press.
17. Miller, G. A., (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. The Psychological Review, Vol. 63, No. 2, March 1956, 81-97.
18. Moray, N. (1959). Attention in a Dichotic Listening: Affective Cues and the Influence of Instructions. Quarterly Journal of Experimental Psychology, 11, 56-60.
19. Sawhney and Schmandt (2000). Nomadic Radio: Speech and Audio Interaction for Contextual Messaging in Nomadic Environments. ACM Transactions on Computer-Human Interaction, 7(3), 353-383.
20. Schmidt, A., Aidoo, K. A., Takaluoma, A., Tuomela, U., Van Laerhoven, K., and Van de Velde, W. (1999). Advanced Interaction in Context. In Gellersen, H.-W. (ed.) Proceedings of HUC '99, Berlin: Springer-Verlag, 89-100.
21. Tarasewich, P. (2003). The Design of Mobile Commerce Applications: What's Context Got to Do With It? Communications of the ACM, forthcoming.
22. Tarasewich, P. and Dideles, M. (2003). Towards a Framework for Information Needs in Mobile Environments. In F. Partovi (ed.), Proceedings of the Northeast Decision Sciences Institute 32nd Annual Meeting, 216-218.
23. Treisman, A. (1960). Contextual Cues in Dichotic Listening. Quarterly Journal of Experimental Psychology, 12, 242-248.
24. Van Dantzich, M., Robbins, D., Horvitz, E., Czerwinski, M. (2002). Scope: Providing Awareness of Multiple Notifications at a Glance. In Proceedings of AVI 2002.
25. Ware, C. and Beatty, J. C. (1985). Using Colour as a Tool in Discrete Data Analysis. Technical Report CS-85-21, Computer Science Dept., Univ. of Waterloo, Waterloo, Ont., Canada, Aug. 1985.
26. Wickens, C. D. and Carswell, C. M., (1997). Information Processing. In Salvendy, G. (ed.), Handbook of Human Factors and Ergonomics, 2nd Edition, New York: Wiley Interscience, pp. 89-129.
27. Wisneski (1999). The Design of Personal Ambient Displays. M.Sc. Thesis, May 1999, MIT Media Laboratory, Cambridge, MA.

Author Index