Supporting early-stage ubicomp experimentation

by

Scott Alan Carter

B.S. (University of New Mexico) 2000
M.S. (University of California, Berkeley) 2004

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy
in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor John Canny, Chair
Assistant Professor Jennifer Mankoff
Professor Peter Lyman
Assistant Professor Maneesh Agrawala

Spring 2007

The dissertation of Scott Alan Carter is approved.

Chair Date

Date

Date

Date

University of California, Berkeley

Spring 2007

Supporting early-stage ubicomp experimentation

Abstract

Supporting early-stage ubicomp experimentation

by

Scott Alan Carter

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor John Canny, Chair

The thesis of this dissertation is that tools and techniques that leverage pre-existing infrastructures, human operators, and the increasing power of mobile devices can improve innovation, development, and evaluation of ubicomp applications. Specifically, we can encourage innovation by developing methods and tools that support situated needfinding; facilitate prototyping by developing tools and methods that eliminate the need to write code by using human operators and simple, rule-based systems; and reduce the burden of evaluations by minimizing infrastructure deployment, capturing and encouraging feedback from critical events, and using devices and interfaces with which participants are already comfortable.

In this thesis, we describe fieldwork that led to a set of core challenges for ubicomp experimentation. From this fieldwork we derive a set of requirements to support early-stage ubicomp experimentation. We then describe the iterative development of a system we built based on the requirements, Momento.

Professor John Canny
Dissertation Committee Chair

Dedicated to my parents, and in memory of Glen W. Coleman.

Contents

List of Figures

List of Tables

Acknowledgements

Jennifer Mankoff was my primary advisor throughout graduate school and influenced every aspect this thesis. I am unspeakably thankful for the help she has given me both professionally and personally.

Anind Dey was instrumental to this work. He served on the proposal committee and provided considerable advice and support.

This work could not have happened without my colleagues and mentors at Berkeley: Scott Lederer, Christopher Beckmann, Jonathan Hey, Tye Rattenbury, Tara Matthews, Jeffrey Heer, Scott Klemmer, James Lin, Jason Hong, Alan Newberger, David Nguyen, Judd Antin, Joe Hall, Nathan Good, Gary Hsieh, James Landay, Warren Sack, Paul Duguid, Geoffrey Nunberg, Ka-Ping Yee, Kate Everitt, Wai-ling Ho-Ching, Holly Fait, Eric Paulos, Anoop Sinha, Miriam Walker, Ana Chang, Yang Li, Sarah Waterson, Matthew Kam, Jack Li, Christine Ro, Mimi Yang, Dan Perkel, Christo Sims, Ryan Aipperspach, and Jaspal Sandhu.

Thanks also to Heather Horst, Laura Dabbish, James Fogarty, Janette Fong, Carol Pai, Amy Hurst, Joe Tullio, Scott Davidoff, Karen Tang, Scott Hudson, Susan Fussell, Peter Scupelli, Christine Ryu, Dahlia Block, Faustinus Gozali, Elizabeth Churchill, Maribeth Back, and Patrick Goddi for their help with this work.

Thanks to Barak Pearlmutter and Hal McAlister for introducing me to research. Thanks to Steve Sigur for introducing me to HCI. Thanks to Theis Dinesen for introducing me to good taste.

Siobhan, a ghrá, go raibh maith agat.

Curriculum Vitæ

Scott Alan Carter

Education

University of California, Berkeley, Berkeley, CA

Ph.D. in Computer Science — 2007

M.S. in Computer Science — 2004

Master's project: *hebb: An Infrastructure for Conveying Shared Interests*

Denmark Design School, Copenhagen

Coursework in digital design — Summer 2001

University of New Mexico, Albuquerque, NM.

B.S. *summa cum laude* with departmental honors in Computer Science — 2000

Thesis title: *Imaging Brain Activation: The MEG Scalp Projection Tool.*

Research interests

My thesis work is Momento, a set of tools that support needfinding, rapid prototyping, and evaluation of mobile or situated ubicomp applications. Momento includes two main components: a desktop interface for researchers and a set of clients for endusers. Researchers can configure the desktop interface to run experience-sampling (ESM) studies or diary studies and to prototype and field test mobile applications that rely on Wizard-of-Oz interaction or event-triggered data retrieval. Researchers can also use the desktop interface to monitor events and carry out actions as wizards. Momento's clients can include SMS and MMS enabled mobile phones, other applications, and a mobile platform that can be configured to support multi-user, multi-week ESM studies, diary studies, or deployment and testing of a prototype application.

I have also published papers on weblog analysis; adaptable interfaces for people with impairments; annotation on public interactive displays; peripheral display theory, prototyping, and evaluation; workplace communication and collaboration; privacy issues in ubiquitous computing; museum guide systems; video conferencing systems; sports innovation; brain imaging.

Journal articles

Carter, S., J. Mankoff, S. Klemmer, and T. Matthews. Exiting the cleanroom: On ecological validity and ubiquitous computing. *Journal of Human-Computer Interaction,* In press.

Matthews, T., T. Rattenbury, and S. Carter. Defining, Designing, and Evaluating Peripheral Displays: An Analysis Using Activity Theory. *Journal of Human-Computer Interaction,* 22(1/2), 2007.

Carter, S. and J. Mankoff. Prototypes in the Wild: Lessons Learned from Evaluating Three Ubicomp Systems. *IEEE Pervasive Computing,* 4(4). Page 51. 2005.

Carter, S., J. Mankoff, and P. Goddi. Building Connections among Loosely Coupled Groups: Hebb's Rule at Work. *Journal of Computer-Supported Cooperative Work,* Volume 13, Issue 3. Pages: 305–327. 2004.

Tang, A., B. Pearlmutter, M. Zibulevsky, and S. Carter. Blind source separation of multichannel neuromagnetic responses. *Computational Neuroscience,* 1999. Proceedings published in *Neurocomputing,* 32–33. Pages: 1115–1120. 2000.

Conference papers (full)

Carter, S., J. Mankoff, and J. Heer. Support for Situated Ubicomp Experimentation. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI),* To appear. 2007.

Carter, S., A. Hurst, J. Mankoff, and J. Li. Dynamically Adapting GUIs to Diverse Input Devices. *Proceedings of The ACM Conference on Computers and Accessibility (ASSETS),* Pages: 63–70. 2006.

Matthews, T., S. Carter, C. Pai, J. Fong, and J. Mankoff. Scribe4Me: Evaluating a Mobile Sound Transcription Tool for the Deaf. *Proceedings of The International Conference on Ubiquitous Computing (Ubicomp),* Pages:159–167. 2006.

Carter, S. and J. Mankoff. When Participants Do the Capturing: The Role of Media in Diary Studies. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI),* Pages: 899–908. 2005.

Matthews, T., A. Dey, J. Mankoff, S. Carter, and T. Rattenbury. A toolkit for managing user attention in peripheral displays. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST),* Pages: 247–256. 2004.

Dey, A., J. Mankoff, G. Abowd, and S. Carter. Distributed mediation of ambiguous context in aware environments. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST),* Pages: 121–130. 2002.

Conference papers (short)

Carter, S. The Role of the Author in Topical Blogs. *Extended abstracts of the ACM Conference on Human Factors in Computing Systems (CHI),* Pages: 1256–1259. 2005.

Carter, S., E. Churchill, L. Denoue, J. Helfman, and L. Nelson. Digital graffiti: public annotation of multimedia content. *Extended abstracts of the ACM Conference on Human Factors in Computing Systems (CHI),* Pages: 1207–1210. 2004.

Workshop papers

Carter, S. Supporting the autobiographical experience of place. *International Conference on Ubiquitous Computing (Ubicomp) workshop: Exurban Noir,* 2006.

Davidoff, S., Carter, S., and Mankoff, J. Can Early-Stage Tools and Techniques for Iterative Design Help Researchers Understand a Problem Space? *International Conference on Pervasive Computing (Pervasive) workshop: What makes for good application-led research in ubiquitous computing?,* 2005.

Mankoff, J. and Carter, S. Crossing Qualitative and Quantitative Evaluation in the Domain of Ubiquitous Computing. *ACM Conference on Human Factors in Computing Systems (CHI) workshop: Usage analysis, Combining Logging and Qualitative Methods,* 2005.

Carter, S., J. Mankoff, and P. Goddi. Representing and supporting action on buried relationships in smart environments. *ACM Conference on Computer Supported Cooperative Work (CSCW) workshop: Public, Community, and Situated Displays,* 2002.

Demos

Carter, S., E. Churchill, L. Denoue, J. Helfman, and P. Murphy. Palimpsests on public view. Demonstration, *International Conference on Ubiquitous Computing (Ubicomp),* 2003.

Carter, S. !hobgoblin. Demonstration, *Museums and the Web (MW),* 2002.

Thesis reports

Carter, S. The design of hebb, a peripheral system supporting awareness and communication, and a study of its impact on small, distributed groups. *MS Thesis, Computer Science Division, University of California, Berkeley,* 2004.

Carter, S. Imaging brain activation: the MEG scalp projection tool. *Undergraduate Honors Thesis, Computer Science Department, University of New Mexico,* 2000.

Technical reports and proposals

Eismann, E. and S. Carter. IVSI: ubiquitous exposure. *Technical Report EECS-2006-96, EECS Department, University of California, Berkeley,* 2006.

Carter, S. and S. Lederer. Informal Videoconferencing and Awareness. *Technical Report EECS-2006-97, EECS Department, University of California, Berkeley,* 2006.

Carter, S. Techniques and tools for field-based early-stage study and iteration of Ubicomp applications. *Dissertation proposal,* 2005.

Abstracts

Hey, J. and Carter, S. Perfect Practice Makes Perfect: The Memory Tennis Accuracy Feedback System. *IEEE Pervasive Computing,* 4(3). Page 54. 2005.

Carter, S., A. Tang, B. Pearlmutter, L. Anderson, C. Aine, and R. Christner. Coactivation of visual and auditory pathways induces changes in the timing of evoked responses in populations of neuron: an MEG study. *Society for Neuroscience Abstracts,* 561.4, Poster, 2000.

Tang, A.C., S.A. Carter, B.A. Pearlmutter, N.A. Malaszenko, L.K. Anderson, C.J. Aine, and R. Christner. Rapid Modification of Populational Neuronal Response Onset Times Via Hebbian Learning: a Non-Invasive Single-Trial Analysis of MEG Data. *Society for Neuroscience Abstracts,* 123C, Poster, 2000.

Patents

Nelson, L. D., L. Denoue, J. Helfman, E. F. Churchill, S. Carter, and P. Murphy. (2005) Methods and systems for extending existing user interfaces. Patent #10/739,186 filed December 19, 2003.

Coding experience

Desktop
Java (extensive), C/C++

Web
PHP, MySQL, Java servlets, Javascript/DHTML/XML, Perl, Ruby on Rails

Mobile
J2ME, .NET Compact Framework, Embedded Visual Basic, Mobile Processing, Python for Series 60, WAP

Work experience

Research assistant, University of California, Berkeley — 2001-2007
Worked on various projects with Prof. Jennifer Mankoff at UC Berkeley and Carnegie Mellon University.

Intern, Fuji-Xerox Palo Alto Laboratory (FXPAL) — Summer 2003
Worked on the Digital Graffiti project with the Social Computing Group.

Intern, Hewlett-Packard Labs — Summer 2002
Worked on the hebb project with the Nomadic Computing Group.

Research assistant, University of New Mexico — 1999-2001
Aided Drs. Pearlmutter and Tang of the UNM CS and Psy-

chology Department with brain imaging research projects in the Brain and Computation Lab at UNM.

General partner, Madpickle Design — 2001
Partner of this graphic and web design company.

Contractor, National Foundation for Functional Brain Imaging — 1999-2000
Provided web design services.

Contractor, University of New Mexico Psychology Department — Summer 1999
Provided web design services.

Web master, New Mexico Daily Lobo — 1998-1999
Created, designed and managed the New Mexico Daily Lobo website.

Production assistant, New Mexico Daily Lobo — 1998
Production Assistant and Assistant Production Manager for the New Mexico Daily Lobo.

Web editor, CNN — Summer 1998
Synthesized individual web pages as web editor for CNN Interactive.

Research assistant, Georgia State University — Summer 1996 and Summer 1997
Programmed in C, planned and created web pages and provided scientific image manipulation assistance.

Teaching experience

Teaching assistant, University of California, Berkeley — Fall 2003
Head teaching assistant for User Interface Design, Prototyping and Evaluation (CS 160)

Professional activities

Program committee member

International Conference on Weblogs and Social Media — 2007

AAAI Spring Symposium: Computational Approaches to Analysing Weblogs — 2006

Reviewer

IEEE Pervasive Computing – 2006

ACM Conference on Human Factors in Computing Systems (CHI) — 2005, 2006, 2007

International Conference on Ubiquitous Computing (Ubicomp) — 2006

Student volunteer

International Conference on Ubiquitous Computing (Ubicomp) — 2002, 2003

ACM Conference on Human Factors in Computing Systems (CHI) — 2002

International Conference on Intelligent User Interfaces (IUI) — 2002

International Conference on Virtual Systems and Multimedia (VSMM) — 2001

Advising

Momento — 2005

Christine Ryu, Dahlia Block, Faustinus K. Gozali, Momento

Input Adapter Tool — 2004

Nick Molchanoff

hebb — 2003

Christine Ro, Mimi Yang

Honors and awards

NSF graduate fellowship honorable mention — 2002

GAANN fellowship — 2001-2002

Best site award for !hobgoblin, Denmark Design School — August 2001

Breece Prize for highest GPA of the graduating class, UNM SOE — 2000

CRA Outstanding Undergraduate honorable mention — 2000

Farris Engineering Memorial Scholarship, UNM — 2000-2001

ITEA Scholarship, UNM — 1999-2000

Outstanding Senior of the Year, UNM CS Dept. — 1999-2000

National Dean's Honor Roll — 1999-2000

Outstanding Junior of the Year, UNM CS Dept. — 1998-1999

Harry Leonard Scholarship, UNM — Fall 1998

Finalist, National Merit Scholarship Program — Spring 1998

Dean's Honor Roll, UNM — Fall 1997-Spring 2000

Amigo Scholarship, UNM — Fall 1997-December 2000

AP Scholar with honor — Fall 1997

Chapter 1

Introduction

In the 15 years since Weiser [175] introduced ubiquitous computing as a goal, the field has made great strides in terms of system building, but with a few notable exceptions (*e.g.*, [3, 128], there has been a dearth of iteration and evaluation. For example, Kjeldskov's and Graham's [95]) review of Mobile HCI systems found field evaluations in only 19 of 102 pieces of published work, and four of those evaluations did not involve working systems. *Real use of real systems is getting short shrift.* For a field to mature, designers and researchers must be able to close the iterative design loop, encompassing both prototyping and evaluation, and learn from their prototypes.

Evaluation can be split into two major stages – *formative*, or before a complete application is built, possibly including *requirements gathering*, before any development at all is done, *early-stage*, in which sketches and rough prototypes may be built; and *summative*, at which point a working system exists. Iterative design may be described as a spiral starting with formative techniques for requirements gathering, sketching and early-stage evaluation, followed by a repeated prototyping and evaluation cycle, in which feedback is obtained on prototypes until a designer is satisfied with the results, followed by implementation and summative evaluation of a working system. Although most evaluation techniques can be used at multiple stages of this cycle, each technique is typically used more at some stages than others. For example, interviewing, ethnography, and other similar techniques are particularly well suited to requirements gathering because they do not require even a prototype system to be effective.

However, while realistic evaluation is necessary for ubicomp experimentation, it can be difficult. As an example of the value of ecological validity, consider the design process of CareNet, an ambient display connecting elders with their families [35]:13. CareNet was deployed in a field experiment that employed Wizard of Oz, an early-stage evaluation technique in which a person (the "wizard") simulates an aspect of the computer's functioning [39, 90, 123] – in this case, activity sensor information. The experimenters found that they needed to "incorporate a daily narrative provided by the drastic life changer [a person who has made major changes to her own life to care for the elder] about how the elder was doing and what her day was like. . . ." This finding arose from participant concerns with replacing the "wizards" with sensors and likely would not have been discovered without the use of an ecologically-valid evaluation. The experimenters make a similar argument about another discovery that arose from their study: ". . . participants got upset when the CareNet Display stopped being ambient. This is the type of problem that *in situ* deployments are good at uncovering." The value of ecologically-valid evaluations is evident in other research systems as well. For example, in a year-long field trial of a system similar to CareNet, the Digital Family Portrait, Rowan and Mynatt [153]:529 found that "behavior shifted gradually with the changes in the seasons." Furthermore, the application required that they install a sensor network in a participant's home. Even though they put considerable effort into planning the deployment, through the evaluation they discovered that their approach to sensor deployment needed iteration and they subsequently developed more robust deployment support.

In summary, ubicomp applications provide broad support for activities by smoothly integrating with the existing artifacts and social structures that are associated with those activities. To achieve this goal, the ubicomp design process must support the development of an understanding of the target environment and extensive iteration in realistic settings. These issues pose an array of challenges, some of which are also faced in other applications. For example, computer-supported cooperative work (CSCW) applications must handle multiple people and potentially multiple locations, and desktop interfaces may use recognition technologies or may need to adjust to different work patterns. But it is the confluence of these different agendas that makes ubicomp especially challenging.

1.1 Characterizing ubiquitous computing

We characterize ubiquitous computing as an approach to designing user experiences that, to use Anderson's [9]:178 phrase, is integrated into the "practical logic of the routine world." Ubicomp applications are designed to address tasks that span the people, artifacts, and places that compose an activity and to address the complex way that activities are interleaved. Ubicomp applications can meet these goals by integrating seamlessly with other successful artifacts and processes. In this way, ubicomp applications can, as Weiser [175]:94) wrote, "weave themselves into the fabric of everyday life until they are indistinguishable from it." For example, although many have lauded the idea that computers will replace paper, in the Myth of the Paperless Office, Sellen and Harper [161] show that users work practices are much more successful, and much more subtle, than a naïve techno-utopian perspective might suggest. Mackay's work with paper flight strips, demonstrates the flexible representation that paper affords, and how users make savvy choices embedded in rich and nuanced work practices [113]. In summary, ubicomp applications that *augment* a user's existing practices can often be more successful than those that seek to supplant them [45, 99].

The term ubiquitous computing has been applied to a broad array of systems; we use the following two-pronged interpretation of ubiquitous computing for the scope of this thesis:

Sensing and Actuation To adapt to changes in activities, ubicomp applications often sense and react to live data about what is going on in the world, or actuate changes in the world around them. As an example, a mobile tour guide may update the information available to the user based on her location (thus reacting to live data), or may help a visitor find the nearest bathroom by causing a light above it to flash (actuation).

Scale Because of the complex and multi-tasking nature of real-world human activity, ubicomp applications often handle one or more of the following complex issues of scale:

Many Tasks Studies have shown that some information workers commonly manage up to 10 basic units of work at a time [55]. Ubicomp applications can benefit from being sensitive to these tasks, or supporting this multi-tasking process. Applications in the sub-area of ubicomp called peripheral displays are often used in multi-tasking situations where the user is monitoring one or more tasks while focusing on others.

Many People Some ubicomp applications must handle issues of collaboration and coordination among groups of people. Examples include shared public displays (*e.g.*, [31]) and systems supporting coordination among small, co-located working groups (*e.g.*, Figure 2.4 [31]).

Many Devices Some ubicomp applications employ multiple devices simultaneously to support a broad array of situations and tasks embedded across time and space. In fact, this epitomizes part of Weiser's original vision of yard-scale, foot-scale, and inch-scale displays.

Many Places Because everyday activities are spread out over both time and space, ubicomp applications often use mobile devices or augment environments. This is the place that ubicomp has most enjoyed broad commercial success, first in the form of smartphones and PDAs, and recently in products that also sense or actuate parts of the user's environment, most commonly providing location-aware services.

The sensing and scale issues of ubicomp make studying these systems more challenging than traditional desktop applications. First, evaluation is hard *to do at all*, making it a difficult process to start for designers whose time and energy is limited. Second, evaluation is hard *to do well*. Even for those who are motivated, there are significant difficulties in conducting ecologically valid evaluations with generalizable results. Ecological validity, by which we mean the extent to which a study matches the actual, "real-world" use of a system, is challenging to achieve because ubicomp applications tend to support not just many aspects of a single activity but potentially also the interaction of multiple activities. It is addressing this challenge that is the focus of this thesis. In this thesis, we focus on evaluation techniques and tools that may be useful in bringing richer ecological validity to ubicomp.

We argue that a nuanced understanding of the particular challenges that arise for ubicomp applications can provide evaluators with valuable advice for how to approach iteration, and can help to identify key research challenges for the future. Some aspects of ubicomp applications, such as basic usability issues, can be evaluated using techniques largely similar to those designed for desktop applications, including discount methods (*e.g.*, [132]) and laboratory studies (*e.g.*, [154]). However, those aspects of applications that depend on an ecologically valid evaluation are particularly difficult to assess. For example, there has been much discussion of the difficulties of building applications at the intersection of computing with groups of people [138, 60, 61, 68], including adoption, sparsity, and critical mass. Without addressing ecological validity, developers risk making and evaluating "a representation without sufficient knowledge of how it actually would work," what Holmquist [72]:50 calls "cargo cult design."

1.2 Thesis statement

The thesis of this dissertation is that tools and techniques that leverage preexisting infrastructures, human operators, and the increasing power of mobile devices can improve researcher's ability to experiment with ubicomp applications in realistic environments. Specifically, we can encourage experimentation by developing evaluation methods and tools that support situated needfinding; facilitate prototyping by developing tools and methods that eliminate the need to write code by using human operators and simple, rule-based systems; and reduce the burden of evaluations by minimizing infrastructure deployment, capturing and encouraging feedback about critical events, and using devices and interfaces with which participants are already comfortable.

1.3 Contributions

Overall, this work offers three contributions. First, we describe interviews with developers in three subfields of ubicomp – mobile applications, peripheral displays, and tangible user interfaces. From this work, and an extensive literature review, we derive five central challenges for situated evaluation of ubicomp – ambiguity and error, sparse data, critical mass, unobtrusiveness, and tool support for realistic environments. Second, we elucidate how the diary study method can be modified to overcome some of these challenges. Specifically, we show that media-based diary studies can help address data sparsity while remaining relatively unobtrusive. From our literature review, interviews, and formative work with the diary study method, we derive a set of requirements for a tool to support realistic ubicomp experiments. Finally, we present Momento which meets these requirements to support a broad range of early-stage ubicomp experiments, from needfinding studies to field evaluations with prototypes.

1.4 Outline

In Chapter 2, we overview evaluation approaches as well as fieldwork by the authors and others with developers in three subfields of ubicomp that, together, flesh out the space of ubicomp applications: peripheral, mobile, and tangible. This work led to a set of core challenges for ubicomp experimentation: ambiguity and error, sparse data, critical mass, unobtrusiveness, and tool support for realistic environments. In Chapter 3 we then describe our work extending the diary study method to meet the needs of realistic ubicomp experimentation. This work led to the development of a technique, involving *in situ* capture and annotation and *ex situ* review, as well as a tool supporting this technique, Reporter. In Chapter 4 we derive from our experiences with both needfinding and prototyping a set of requirements for a tool to support early-stage ubicomp experimentation. We then describe the iterative development of this system, Momento, in Chapters 5 and 6.

Chapter 2

Background

In this chapter we present a literature review and interviews investigating three key areas of ubicomp – peripheral displays, mobile applications, and tangible user interfaces – concentrating on the difficulties encountered in prototyping and evaluating these systems. From these we derive a set of challenges for ubicomp evaluation, and then explore how past work has attempted to address these challenges for different types of evaluations, including needfinding, prototyping, and lab and field studies.[1]

2.1 Field work with ubicomp developers

We conducted interviews with developers in one subfield of ubiquitous computing – mobile systems – and summaries of interviews conducted in two other subfields – tangible user interfaces and peripheral displays. Together, these subfields span the key characteristics of ubicomp. Peripheral displays represent *sensed* information to help people coordinate *multiple tasks*. Mobile applications are designed to be used in *many places* and usually need to work across *many devices*. Tangible interfaces *sense* actions in the physical world and *actuate* responses to them. Furthermore, each of these fields include technologies that support both individual and group tasks. Through fieldwork

[1] This chapter is based on [24].

with researchers who are developing software in an area, we can gain an understanding of the challenges of development as practiced and find opportunities for tool research.

In presenting the findings of our field work below, we concentrate on the difficulties encountered in prototyping and evaluating these systems. Examples of successful evaluations and prototyping are often published: information about problems is far rarer.

One common theme that was expressed by developers in many of our interviews was the need to develop functional prototypes early on that could enable situated, ecologically valid evaluations. For example, two peripheral display designers felt it important to gather longitudinal data, one mobile developer wanted to know how an application "changes [a user's] day," and one tangible developer discussed an interest in understanding failure modes to help drive development of a robust, complete system. Interviewees felt that prototypes in each case could be a *means* of answering questions.

2.1.1 Mobile applications

Mobile applications are those deployed to personal devices that people carry from place to place. (See Figure 2.1 for an example). Mobile applications often must handle issues of scale: they may be expected to function appropriately in *many places* or to work across *many devices*. Many mobile applications are designed to be used collaboratively by two or more people. Mobile devices represent one of the most successful domains of ubicomp: billions of people across the globe use them on a daily basis. Yet we found that building and evaluating applications for mobile devices remains challenging.

Method

We conducted interviews with nine designers of mobile applications. We focused on developers who had deployed applications to personal digital assistants (PDAs) and mobile phones. Six participants held research positions; the other three worked in non-research, industry positions. Three of the participants were primarily designers,

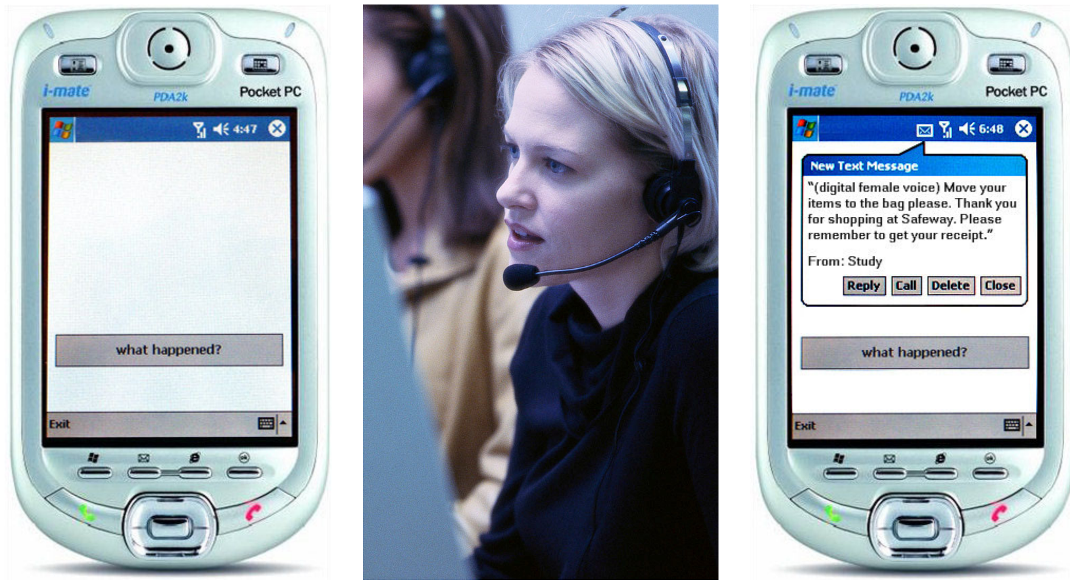


Figure 2.1. Matthews *et al.*'s functional prototype of the Scribe4Me system, which provides an on-demand transcription service for the deaf. By pressing "What happened" the user causes the previous 30 seconds of audio and an image to be sent to a remote wizard who sends back a transcription [120]. In our field study, the MMS network caused delays of three to five minutes.

three were primarily developers, and three were both. Participants had designed between two and four mobile systems over the last one to three years.

Interviews were conducted in person. We asked participants a set of open-ended questions addressing difficulties they encountered designing, building, and evaluating mobile applications.

Results

Interviewees considered ecological validity paramount in evaluations of mobile applications. This issue led them to concentrate on field studies, but they encountered difficulties developing prototypes robust enough for use in uncontrolled settings.

Interviewees believed it vital to understand how mobile systems are used in field settings, but expressed concern that needfinding techniques suitable for desktop settings would not garner results that could translate to real use for mobile applications. One developer commented that "new concepts need to be vetted in the field" before they could be considered valid. Needfinding techniques suitable for gathering situated data, such as diary research, were seen as suitable solutions. Still, developers cited "staying on top of users" during the study in addition to lengthy perceived set-up time as reasons why they were not inclined to run such studies. These are challenges common to non-mobile designs as well, and ones that should be overcome to promote needfinding.

Our interviews verified what Kjeldskov and Graham [95] suggested in their review of published mobile HCI research: many mobile developers relied on existing knowledge and trial and error to derive new designs. They also point out that many developers conduct extensive studies of mobile use that represented research contributions in their own right. We did not see this phenomenon in our interviews, but there are several reports in the literature of more extensive studies conducted by ethnographers working closely with designers that variously included extended participant observations and interviews and analysis of collected data [75, 178] as well as diary studies [143]. For example, Horst [75] conducted an anthropological investigation of cell phone use among low-income Jamaicans over a one-year period, finding that people use cell phones to keep alive essential social network connections. Woodruff *et al.* [178] lived with teenagers for a one-week period to understand how they use push-to-talk technologies, from which they found inspiration and design goals for a social audio space.

In the transition from needfinding to evaluation, interviewees rarely used lightweight prototypes. This trend arose because developers strongly believed that it was important to test their tools in realistic settings, but that it was difficult to contrive realism using lightweight prototypes. Instead the developers concentrated on mock-ups of their displays that they then used in cognitive walkthroughs (similar methods, such as heuristic walkthroughs, have also been used in the literature, *e.g.*, [94]). Using this approach, developers could "find the really big and the really small" problems with the design without worrying about "trying to get the user to imagine" that they are in a realistic situation during a study.

Interviewees used a variety of different mobile development platforms once they were ready to create full prototypes, but all reported difficulties, especially when attempting to deploy their application to more than one type of device and across different infrastructures. For example, one participant commented, "what was a shock to me was to learn that lots of the Java JSR specs [mobile APIs] are optional. So different operators and – no worse than that – different devices might implement one function but not another or implement it a different way. . . ." Another participant lamented that different cellular networks operate differently enough that sometimes "you have to make versions for different models and networks, which. . . explodes the development branch tree."

Two interviewees used controlled lab studies to evaluate interaction issues. However, ecological validity was a lesser concern in these studies; the developers concentrated on the user's ability to "[get] from A to B" in the interface. In their review of mobile evaluations, Kjeldskov and Graham [95] show that this use of controlled studies is common. Using this approach, they were able to find critical interaction problems – for example, that screens were too cluttered to be interpretable. But interviewees did not believe that the studies were useful ways to identify problems more related to actual experience – for example, the level of navigation complexity that users were willing to tolerate.

All nine interviewees had conducted a field experiment. One commented that, "I think the main thing we want to know is how [the application] actually affects what they do. . . how that information changes their day," and developers considered field experiments the only reliable way to find that information. However, they did report a number of issues that stood in the way of conducting field experiments. In addition to the challenges with developing functional prototypes described above, because of the plethora of different mobile operators, plans, and devices, developers had difficulties planning studies. Mobile operators, in particular, were a concern, "sometimes they will change something during the study. . . and your [application] will not work any more or you will have a different payment plan," and "sometimes it is hard to find out what [the operator]'s limits for various features. . . like data limits on messages." As an example, the Scribe4Me system [120], which sends audio and photographs across the MMS network to provide translations for the deaf (see Figure 2.1), we encountered rare delays of up to nine hours when messages had to cross between service providers on rare occasions.

Interviewees often had trouble gathering data in their field experiments because the activities their applications augmented occurred infrequently. For example, a researcher testing a transit application found that most participants used the device only twice a day – to and from work. The researcher felt that to gather enough data to guide the next iteration the deployment would need to run for months, and "you either have to build something robust enough to last, which takes a long time, or keep fixing it when it breaks, which also takes a long time... and is frustrating."

Once the pragmatic concerns of deploying technology is overcome, developers encounter evaluation challenges similar to those in needfinding studies. For example, in their study of a mobile presence awareness device for ski instructors, Weilenmann *et al.* [174] found that "the observer's task is difficult – it is simply not possible to be everywhere at the same time" and used participant observations and focus groups to evaluate the tool. The developers we interviewed had similar concerns and chose to run either diary studies or to rely primarily on interaction logs.

Discussion

Ecological validity was a primary concern among mobile developers, as a way both of vetting new concepts and seeing the effect of an application on "what they do. . . how [it] changes their day." Furthermore, developers felt that field experiments were a good way of addressing this concern. Intuitively, this makes sense – precisely what makes an application mobile is that it is used in many different situations. However, especially when clean, generalizable results are desired, conducting field experiments is challenging due to a variety of development, methodological, and pragmatic difficulties. Controlled studies represent an alternative, and attempts to address ecological validity in controlled experiments have proven valuable, though they may be limited to applications that are mobile only within a limited environment.

Our participants verbalized a concern about the difficulty of collecting ecologically valid data with lightweight mobile prototypes. Others have reported similar concerns. For example, Rudström *et al.* [155], in a paper prototype study of a mobile social application, found that participants had difficulty reflecting upon how their use of the application would change if they actually were mobile and using an interactive system. We [27] also ran a similar paper study of the interaction between a mobile device and a public display. However, the task required participants to act as though they had serendipitously encountered the display, which was difficult for them to enact.

With heavyweight prototypes, interviewees often employed controlled studies, typically in lab settings, because these studies are more forgiving of the fragility of early-stage technology, and because data across participants can be more easily compared. However, the interviewees were concerned that the contrived nature of such studies limits their ecological validity. Oulasvirta *et al.* [140] articulated an important shortcoming of lab studies in the mobile domain is that the attentional demands of mobile applications cannot be simulated in lab environments, because in realistic environments a plethora of activities interact to constrain severely the continuous periods that participants can attend to mobile devices.

To address this, a few researchers have taken steps to make controlled studies more realistic and also to devise more rapidly buildable approximations of a system that can be used to move controlled studies into the field. Kjeldskov *et al.* [96] recreated a hospital situation in a lab and ran controlled experiments in which participants had to move and interact with other devices to complete tasks. They showed that they were able to find all of the usability errors in their lab evaluation that they found in a field evaluation of the same prototype. Kjeldskov and Stage [97] also ran controlled studies that integrated the varying body movement and attentional demands that would be present in mobile situations. In Yeh *et al.*'s [179] controlled field experiment with 14 biologists of the ButterflyNet system, a device ensemble comprising a mobile device and an augmented paper notebook, the insight that enabled this work to take place in a tractable fashion was to use a handheld Windows XP machine to simulate the features of a future digital camera.

Because of the high time investment and development costs of classic field observation and high-fidelity deployment, researchers have recently begun to explore techniques that can provide sufficiently rich data at lower cost. For example, researchers are increasingly using diary and experience sampling studies to provide design guidelines for mobile applications. Okabe and Ito [136] used interviews and diary studies to learn how people use mobile phone picture technologies, showing that personal archiving and maintaining distributed-copresence are common uses. In their article examining text messaging amongst teenagers, Grinter *et al.* [59]:442 talk about using diary studies because direct observation "would be impractical" and "teenagers were hesitant about being directly observed." Palen *et al.* [143] used a voice-based diary to study mobile phone calls, finding design issues with public mobile phone use. Also, the PlaceLab group at Intel Research Seattle ran an experience-sampling study to understand how factors such as activity and mood effect location disclosure in mobile applications and used this data in the design of a social location disclosure service application [36, 166]. Finally, Abowd *et al.* [4] introduced the notion of a *paratype*, a modified diary study in which experimenters first describe the proposed functionality of a tool to participants and then ask participants to diary situations in which they believe that tool would be useful.

To conduct field studies, developers reported having to develop prototypes for multiple different platforms. The difficulty of deploying multiple different versions of a tool to meet different environmental demands (*e.g.*, developing different Web pages for Internet Explorer and for Mozilla) is not new. However, as one developer suggested, this problem "explodes" when each device and network has different demands. New prototyping tools, such as Python for Nokia Series 60 phones [133] or Mobile Processing [106] can reduce iteration time, but are still limited in device support and do not address differences in network support.

After deploying a technology, developers encounter evaluation challenges similar to those in needfinding studies. Similar solutions (such as diary research) can be used, and augmented with logs of system use. For example, some researchers have relied primarily on video and interaction logs to evaluate field deployments [51, 15].

2.1.2 Peripheral Displays

Peripheral displays are tools that enable quick and easy access to information. Though extensive use may engender enough familiarity to enable many information sources to be peripheral, peripheral displays are specifically designed to be glanceable and non-interruptive. These displays are often used in ubicomp because their glanceability enables them to scale across *many activities* so that people can monitor many information streams outside of their focal activity, while their non-interruptive nature minimizes the extent to which they distract from that activity. An example of a peripheral display is Pinwheels, which mapped the spin of pinwheels to the rate of change of a variety of information sources.

Matthews [119] found that the central problem facing developers of peripheral displays is that metrics for success are not well defined. One participant summarized this issue saying that while "most technology that is out there is about maximizing efficiency" that is often not the case with peripheral displays, causing designers to "reevaluate [standard] systems of evaluation."

Broadly speaking, peripheral displays require a different style of technological intervention than traditional 'foreground-based' user interfaces. As such, it may be challenging to precisely specify the most appropriate metrics for success and to discover appropriate interventions. Needfinding is used to address this issue because it helps researchers understand the specific context in which a display will be used. Researchers have found sketches effective in needfinding studies to facilitate concrete comparisons between different designs and to help participants express their expectations for a display. Matthews *et al.* [118] conducted needfinding interviews and sketch studies that led to the IC2Hear sound awareness display. In this study, the sketches gave users semi-concrete display ideas to discuss. The rough nature of the sketches encouraged critiques and suggestions, improving the prototypes created based on interview results. Similarly, Sengers *et al.* [162]:54 instructed participants to "reflect on aspects of their current relationship and technology use within that relationship, and [had] them sketch novel designs for communication devices for couples to use."

Researchers also have begun to derive metrics and design guidelines for peripheral displays. Mankoff *et al.* [114] adapted heuristic evaluation to ambient displays, a subset of peripheral displays that focus on aesthetics and tend to convey information of low criticality. Those heuristics encode design goals for peripheral displays. McCrickard *et al.* [124] presented IRC, a design model for classifying different types of peripheral awareness systems along the dimensions of interruption, reaction, and comprehension. The model can be used analytically to understand how a design might affect the user along those dimensions. It can also guide empirical evaluations by helping to identify relevant metrics. Finally, Matthews *et al.* [122] presented a set of evaluation metrics and guidelines derived from past literature and a user-centered, activity theory framework. Metrics include appeal, learnability, awareness, effects of breakdowns, and distraction. Guidelines focus on prioritizing metrics depending on design dimensions identified as part of the framework.

Shami *et al.* [163] developed Context of Use Evaluation of Peripheral Displays (CUEPD), an evaluation method that relies on active user participation and emphasizes the experience of using peripheral displays. CUEPD captures the context of use through user scenario building, enactment, and reflection. Designers can use CUEPD once they have a working prototype to improve future designs. This new method attempts to increase realism in a laboratory experiment with scenarios collaboratively created by the designer and user. It also provides guidance for evaluation metrics by suggesting survey question categories: noticeability, comprehension, relevance, division of attention, and engagement.

Peripheral display developers have leveraged multiple research toolkits. Because peripheral displays often employ physical user interface elements as their display modality, developers have benefited from recent research on tool support for physical interaction design, including Phidgets [56], iStuff [10] and d.tools [65]. Furthermore, Matthews' *et al.*'s [117] Peripheral Display Toolkit, based on requirements derived from these interviews, has helped to structure the creation of functional prototypes.

Most controlled studies and field evaluations of peripheral displays have focused on issues such as usability, awareness, and distraction. For example, the Scope interface was studied in a pilot lab study to identify major usability problems and to drive design iteration [172]. Participants were asked to perform tasks that involved interpreting the interface. Data included the time to complete tasks on the Scope and subjective usability ratings from a survey of Likert-scale questions. Ho-ching *et al.* [70] compared the awareness provided and distraction caused by two peripheral displays of sound in a dual-task lab study. In a multiple-task lab study, Matthews *et al.* [121] compared the multitasking efficiency benefits caused by a peripheral display using various abstraction techniques. Data included time to complete tasks (indicates task flow and distraction), time to resume a paused task after a new update (indicates awareness), number of tasks and window switches (indicates awareness) and user satisfaction.

The iterative design of Sideshow, a peripheral display by Cadiz *et al.* [22] was particularly successful in improving the display based on user feedback. For example, laptop users requested an "offline" mode that showed stale data. Though hesitant to show outdated information, designers added this feature and got positive feedback from users. Many such iterations improved Sideshow's usefulness to users. This successful iteration process was facilitated in large part by a focus on making Sideshow easy to maintain and update. During a 9-month period, 22 new versions of Sideshow, a graphical peripheral display of various information streams (*e.g.*, meetings, email, IM, co-worker presence, traffic, weather), were released with bug fixes and new features. The updates were made based on a constant dialog with users, who submitted bug reports and email feedback. Sideshow had an advantage over other ubicomp applications, though, being a software program running on a desktop computer. Off-the-desktop applications are more difficult to update, making frequent modifications less practical.

2.1.3 Tangible User Interfaces

A primary goal of ubiquitous computing is the creation of systems that augment the physical world by integrating digital information with everyday physical objects. They typically *sense* and/or *actuate* aspects of the world. The art of designing these interfaces involves leveraging the unique strengths that the physical and electronic worlds have to offer, rather than naïvely replicating the interaction models of one paradigm in the other. For example, in Mackay's work with paper flight strips, the most useful design was one that augmented existing paper flight strips rather than replacing them entirely, combining the flexibility of paper with the speed of digital capture and presentation.

Klemmer [98] found that the extensive expertise needed to build robust tangible interfaces presented the largest challenge to evaluation for interviewees. For example, in each of the three projects that employed computer vision, the team included a vision expert. Even with an expert, writing vision code proved challenging. Writing code without the help of a toolkit yielded applications that were unreliable, brittle, or both.

In addition to Mackay *et al.* 's [112] fieldwork with air traffic controllers, other researchers have conducted needfinding studies of tangible interfaces that successfully translated to prototypes. In their study of web designers, Newman *et al.* [131] found that designers used several different representations of Web sites as they worked, allowing them to concentrate on different aspects of design. This work led to tools supporting these different aspects of design, including Designers' Outpost. Also, Yeh's [179] fieldwork led to the creation of tools to support data capture for biologists working in the field.

Prototyping was beneficial to interviewees. Klemmer's results demonstrate that the interviewees' prototypes helped them better understand the problem their tool was trying to solve, and that the interviewees different approaches provided different insights. Klemmer also found that the heterogeneity of ubicomp's input technologies may require different support architectures than GUI toolkits provide. The challenges of this heterogeneity and the benefits of toolkit support for managing both input and presentation suggest that user interface management systems (UIMS) may be useful for ubicomp [69]. Furthermore, a significant difficulty in program debugging is the limited visibility of application behavior [41]. The novel hardware used in tangible interfaces, and the algorithmic complexity of computer vision, only exacerbate this problem.

Researchers have conducted a handful of controlled studies of tangible interfaces. Klemmer [101] evaluated Outpost with professional web designers. Participants were asked to "speak aloud" about their experiences while they completed an information architecture design task. Also, Fitzmaurice *et al.* [50] implemented and evaluated a tangible interface to Alias Studio, a high-end 3D modeling and animation program. The evaluation found that users rapidly learned how to perform complex operations. Finally, McGee *et al.* [125] conducted an evaluation comparing traditional paper tools to Rasa, a system that extends tools currently used in military command post settings with a touch-sensitive smart Board, gesture recognition on ink strokes written on the Post-it notes, and speech recognition on verbal commands. The researchers took the novel step of shutting down the system halfway through the experiment to evaluate users' response to breakdowns.

Extended field deployments of tangible interfaces are rare, but some evidence shows that they can yield important insights. Lee *et al.* [105] conducted a longitudinal study of an augmented paper interface for student design teams. Specifically, they deployed iDeas, a system that leverages digital pens and cameras to support design practice. They deployed the system for two academic quarters with fifty-eight design students and recorded over four thousand pages of authored content. Their results showed that their tool enabled new behaviors, including reflection upon design process. Improved prototyping tools and evaluation methods have the ability to lower the threshold for such valuable deployments.

2.2 Challenges Iterating Ubicomp Applications

This formative work revealed that designers of ubicomp applications struggle with ecological validity throughout the design process. For example, Figure 2.2a shows a system that spans mobile and public applications to sense and display awareness information. This system was difficult to prototype because it spanned devices, places, and users, and it was difficult to evaluate because most important events (*e.g.*, impromptu meetings similar to the one pictured) occurred spontaneously. Our interviews and our literature survey, along with case studies described in [25], suggest that there are five particularly salient ways that the sensing and scale of ubicomp resist easy *prototyping* and *ecologically valid evaluation*: handling ambiguities and error, dealing with sparse data, reaching critical mass, remaining unobtrusive, and developing tools for realistic environments.

Ambiguity and error: Ubicomp applications that depend on sensed data and associated inferencing technologies must mitigate ambiguity and error, a process that necessarily involves the end user and thus must be reflected in the evaluation process. Bellotti *et al.* [14] discuss some of the issues that arise from inferencing, including recovery from mistakes, clearly articulating the target of a command, and telling if the system is attending. Approaches addressing these core usability issues, such as input repetition and choice [115], can only be tested if recognition and ambiguity are included in evaluation in a representative fashion. Additionally, the accuracy of any sensing and inferencing has a

Figure 2.2. The awareness prototype deployed in a field setting. Location and availability of users were sensed through users' mobile devices and Wizard of Oz input. The public displays relied on three different research prototyping systems [28].

(a) (b)

Figure 2.3. **(a)** Our system designed to encourage communication and collaboration among work colleagues. Pictured here are two components of the system: an interactive public display and beneath it a badge reader. The value of the system was directly related to the number of participants actively using it [27]. **(b)**, Matthews *et al.*'s toolkit facilitates the control of peripheral devices such as this orb from Ambient Devices, which can unobtrusively change color and pulse to indicate different information patterns [117].

Figure 2.4. Churchill *et al.*'s Plasma Poster was an interactive public display designed to encourage informal content sharing and conversations. The system was designed for informal social situations, such as a café (pictured here), which are difficult to recreate in lab settings [31].

huge impact on the outcome of such an evaluation, and it may be difficult to prototype accurate sensing and inferencing systems.

Sparse data: Some tasks may naturally occur only occasionally (such as commuting to and from work), or may be difficult to sense (such as an emotional response). This impacts prototyping because prototypes must function in the myriad settings where tasks may occur and because data collection for sensing purposes may be difficult. For example, any system that depends on a large corpus of labels, sensed data for inferencing will be especially difficult to prototype if data is sparse. Overcoming this challenge often requires running evaluations over large amounts of time, people, or places.

Critical mass: For ubicomp applications that must scale to involve many tasks, places, people, or devices, reaching critical mass along the relevant dimension is important to ecological validity. This requires prototypes to robustly scale. It affects evaluation because difficulties such as adoption by many people (such as in Figure 2.3a) or unanticipated interference with existing activities may arise. Also, these tendencies suggest that a realistic use scenario for a ubicomp application includes not only the people, artifacts, and places involved in a single target activity, but potentially also other activities in which each target person or group, artifact, or place is involved.

Unobtrusiveness: Monitoring the use of any application can change user behavior. For conventional applications, the effect of monitoring is usually small enough not to impact an evaluation. But ubicomp applications may have only subtle effects on behavior, and the effects of monitoring may therefore interfere with an evaluation's outcomes. Additionally, prototypes themselves often have properties that may make them stand out. To be unobtrusive, prototypes work best when they are refined, of appropriate size and weight and requiring only appropriate amounts of attention (such as the Ambient display in Figure 2.3b). This makes evaluation at the early stages of design particularly challenging: Consider three prototypes which differ significantly in terms of size, weight and functionality. Should a developer invest more time to make prototypes more appropriate before testing them? If not, can she trust the results of her tests? Even when prototypes exhibit subtlety, evaluations must leverage subtle techniques that provide data without causing major changes in use.

Tool support for realistic environments: We take the research goal of ubicomp to be systems that integrate into "the practical logic of the routine world" [9]:178. This raises two issues. The first is that building systems that operate in the everyday world – even one-off prototypes – is difficult and time-consuming. For example, Wizard of Oz prototypes are excellent for early lab studies, but do not scale to longitudinal deployment because of the labor commitment for human-in-the-loop systems. The second is that, even if the system works, it can be difficult to build tools to capture and analyze the longitudinal user experience of a system in the real world. Consider the rich context of use of the interface in Figure 2.4. Video recordings and system logs are both helpful, but the traditional methods of working with this data have often been prohibitively time-consuming. Lighter-weight techniques for dealing with rich capture of longitudinal user data are needed.

2.2.1 Summary

McGrath [126] argues that an evaluation is complete to the extent that it is precise, realistic, and generalizable. His analysis of evaluation methods highlights that controlled evaluations maximize precision, while field studies and experiments maximize realism, and that it is through a combination of these different approaches that designers can arrive at generalizable theories of application use. Thus, developers will likely encounter the challenges we derived above in many different types of evaluations. In the following sections we explore the implications of these challenges for different stages of the iterative design process: needfinding, prototyping, and lab and field evaluations.

2.3 Challenges of needfinding

In this section we review strategies for needfinding, and argue that event-driven field-based needfinding methods best meet the challenges of ubicomp evaluation.

In the past decade, it has become increasingly common for user-centered design efforts to begin with some form of observation-based needfinding. Observation plays a role not only during needfinding but also during field studies and other types of situated evaluation of technological prototypes. This grounds subsequent design discussion in the actual practices of actual users and provides an opportunity to unearth insights that may guide design. Needfinding and observational work ranges from rigorous and labor-intensive methods such as ethnography [64] – comprising intensive qualitative observation that can last multiple years – to more cost-sensitive and applied methods such as contextual design [18, 73]. Returning to our working definition of ubicomp as being computing that is concerned with "the practical logic of the routine world" [9]:178, it becomes clear why qualitative field observation methods have enjoyed some success in user-centered ubiquitous computing efforts (see *e.g.*, [36, 136, 143, 59, 78]).

The primary difficulty with gathering high quality data through observation is remaining *unobtrusive* while monitoring potentially *sparse data*. First, the act of observing often has a "Heisenberg uncertainty principle"-like effect in that observing participant behavior can change it. This is particularly problematic for highly reactive activities such as the monitoring peripheral information. For example, a developer might be interested in how people react to noise in the hallway. By asking a participant directly about the noise, the developer might cause her to notice it more. A more unobtrusive approach might be to observe or videotape the participant and ask about the situation. Second, events of interest to the evaluator may be rare, such as serendipitous encounters. Important events may also be hard to observe. For example, how might one observe interruptability? We suggest that approaches to observation that can handle *sparse data* are important directions forward (*e.g.*, [77]). Lower-cost observational methods that are perfectly appropriate for more constrained settings may run into problems with unobtrusiveness and sparse data. While a carefully structured evaluation can help to mitigate this, evaluators may be forced to reduce realism in the process (for example, by simulating events at a higher frequency than they might otherwise happen in order to observe a participant's response).

When realism is important, an evaluator may turn to situated techniques, with a remote evaluator. This can make it feasible to conduct evaluations over a longer period of time (addressing data sparsity), while the removal of the evaluator can help to make the experiment less obtrusive (although monitoring can interfere as long as the user is involved or aware of data being gathered). Below, we discuss some situated techniques that are especially appropriate for ubicomp because they can provide a balanced solution to the problems of realism, unobtrusiveness, and data sparsity. A primary challenge in observation efforts is that capturing data is often cheap and easy, but that accessing that data later for use as a design resource can be challenging. Interfaces that help manage this data promise to increase the value of observation. For example, designers and anthropologists have used the ButterflyNet system to capture a variety of media in the field and search, manage, and share that data *ex situ* [179].

Other systems facilitate capture of field data in situations in which participants' attention is distributed amongst multiple activities, but these systems usually do not support a range of media nor provide communication support. For example, Cybertracker and systems developed by Pascoe *et al.* allow field workers to track animals by providing a memory prosthesis, but those systems neither require nor provide support for interactions between experimenters and participants [38, 144]. Also, others have developed episodic memory prompts but that were not used for evaluation. Forget-me-not automatically captured event context and displayed icon-based cues for each event [103]. Eldridge *et al.* used video to aid recall but again did not use the system for evaluation purposes [49]. Also, Carmien explored prompts on personal devices for personal coaches for the memory impaired [23].

In the following section, we explore how event-driven field-based needfinding methods can be used to address ubicomp evaluation challenges, especially realism, unobtrusiveness, and data sparsity.

2.3.1 Event-driven field-based needfinding methods

Using event-driven field-based needfinding methods, experimenters can gather realistic data while remaining relatively unobtrusive to participants. In this section, we use five differentiating terminologies to guide our discussion of different event-driven field-based needfinding methods (see Table 2.1: communication between an experimenter and participants (synchronous or asynchronous, *e.g.*, a web-based survey is asynchronous while an interview is synchronous); the location of the experimenter with respect to the participants (local or remote), the interface between participant and experimenter with respect to the event-of-interest (*in situ* versus *ex situ*); who controls the capture of events (experimenter or participant); and what indicates events of importance (time or activity).

	Capture	Indication	Interface	Location	Communication
Time-contingent ESM	experimenter	time	*in situ*	remote	asynchronous
Event-contingent ESM	experimenter	activity	*in situ*	remote	asynchronous
Diary study (feedback)	participant	activity	*in situ*	remote	asynchronous
Diary study (elicitation)	participant	activity	*ex situ*	remote	synchronous
Participant observation	experimenter	activity	*in situ*	local	either
Contextual inquiry (CI)	experimenter	activity	*in situ*	local	synchronous
Cultural probes	participant	activity	*ex situ*	remote	asynchronous
Phone-interview ESM	participant	activity	*in situ*	remote	synchronous
Modified CI	either	activity	*ex situ*	local	synchronous

Table 2.1. Event-driven field-based needfinding methods. Methods are categorized by who controls the capture of events (experimenters or participants), what indicates events of importance (a specific activity or time), when the captured event is probed for deeper understanding through questions or annotations (*in situ* or *ex situ*), where the experimenter is with regards to participants (local or remote), and how experimenters and participants communicate (asynchronously or synchronously). Methods above the horizontal line are standard, while the methods below were extrapolated from the categories but are as yet untested.

Experience Sampling

In the Experience Sampling Method (ESM), participants are interrupted throughout the day to answer a set of questions *in situ* at time- or activity-based intervals specified by a remote experimenter [85, 11]. Participants must respond to a short survey – effectively communicating with experimenters asynchronously. The technique in its classical form is very appropriate for the needs of ubicomp.

By asking questions at a low frequency, and keeping the experimenter remote, the technique can remain fairly unobtrusive. In order to keep the time commitment of participants low, while still capturing information about sparse data, experimenters may want to use a variation of the technique called event-contingent ESM that attempts to ask questions at meaningful times rather than at random times (See [83, 152, 176]) for more information on this technique). Ideally, event-contingent ESM asks questions only at the rare moments when something interesting happens, rather than hoping that question and event will coincide.

While ESM is situated, realism is still a concern for this technique because the remote experimenter may not have rich data about the situations on which the user is reporting. Researchers are beginning to look at media capture as a way of increasing realism (see [13]). Also, while ESM is useful for measuring the amount of time participants spend on everyday tasks, it is not as useful when one is interested in events in which a participant was unable to do something because of a limitation, or when a participant action is difficult to sense.

Diary studies

The diary study is a *participant-driven* method of understanding participant behavior and intent *in situ* that minimizes the effects of observer. Similar to ESM, the experimenter is remote from participants. Diary studies differ from other field study methods in that experimenters are remote from participants and participants control the timing and means of capture. When experimenters are local with respect to participants, as in contextual inquiries, they are able to discuss the implications of events and actions with participants immediately. These studies yield data less pigeonholed by a participant's particular perception of an event but are subject to presentation effects (*i.e.*, participants may act differently because of the presence of the experimenter) and are time-consuming and difficult to scale. Also, when experimenters control capture they are able to obtain objective data about participant's activities but do not necessarily gain an understanding of the events that are important to the participants. An example of such a method is experience sampling (ESM).

Diary studies can be broken down into those that use media captured by participants as prompts for discussion in interviews (elicitation studies) and those that require participants to answer predefined questions about events (feedback studies). Feedback studies may also require participants to capture media to serve as prompts, but the principle difference between elicitation and feedback studies is that elicitation studies involve *synchronous* communication between experimenter and participant (*e.g.*, interviews) while feedback studies involve *asynchronous* communication between experimenter and participant (*e.g.*, questionnaires). In some studies the methods are combined, with results from feedback serving as prompts for discussion during the elicitation study.

Another difference between feedback studies and elicitation studies is that in feedback studies participants should provide information about an event immediately after they perceive it, whereas in elicitation studies participants only capture some aspect of an event when it occurs and provide information about it later during interviews. Thus, a typical feedback study will ask participants to answer questions about some event as soon as it occurs, whereas in an elicitation study participants merely capture some information about the event that will serve as a memory cue during a later interview. Feedback studies have the drawback of potentially overburdening participants with questions, especially when the number of events reported is high [147]. Because participants can rapidly capture prompts, such as a photograph, elicitation studies tend to be much less burdensome. But because questions are asked at the time of the event, or *in situ*, feedback studies are more likely to provide accurate responses to questions that depend on recall of the event. Thus, the two methods represent a tradeoff made between accurate recall but burdensome logging (feedback) versus potentially inaccurate recall but unobtrusive logging (elicitation).

Thus for elicitation studies, capture is quick, but as the captured media still represents a subjective point-of-view, it has some empirical value. Barsalou posited that episodic memory can be improved when a person is presented with cues about an event such as who was involved, where it occurred or what was done just before and after the event [12]. However, while experimenters have recently begun using diary studies using photo-elicitation, it is not evident how well media capture these cues and to what extent media facilitate participant reconstruction of events. Also, different media *types* will likely evoke different reconstructions and attitudes towards an event, but no study has yet shown how.

Review of studies run by other experimenters Feedback studies usually rely on paper-based forms as the feedback medium [33, 160, 151, 58]. However, one of the concerns with using paper-based feedback studies is sustained subject participation. In an attempt to address this issue, Palen and Salzman experimented with cell phones as a feedback medium [142]. From their studies they derived ways to encourage subject participation, including the use of periodic reminders and reimbursement strategies. They also recommend that experimenters provide participants with feedback about the level of detail of their responses.

Researchers in both behavioral and technological research communities have only recently begun to explore elicitation studies. Elicitation studies are rare with the exception of photo-elicitation studies. Brown *et al.* used photo-elicitation to understand design requirements for information capture devices, and O'Hara *et al.* used the same method to understand transaction decisions [20, 134]. In his study of young Buddhist monks in Sri Lanka, Samuels compared photo-elicitation to word-only interviews and found that participants were far more detailed in their description of everyday events with photo elicitation [157]. According to participants, the difference arose because the photos that they took had more "meaning and value" to them and that they could "explain more when the pictures are close at hand." Samuels also found with photo-elicitation that participants were better able to make novel associations among tasks and that participants tended to remain more focused on the interview.

Clark-Ibáñez also used photo-elicitation in her studies of children attending elementary schools in urban environments [32]. She described that background information in photos can often be crucially important, specifically citing a case in which participants discussed the "'tagging' of gang names and symbols" in the periphery of one image that she did not initially notice. She also found that photos tended to hold participants' attention and found them useful for structuring interviews. However, as she developed the photos herself, participants could not review and potentially remove photos before the elicitation interview, frustrating some.

Sampson-Cordle used photo-elicitation to construct photo essays, or combinations of interview transcripts and photos about a related topic, in her study of a small rural school and the community in which it was situated [156]. She found that participants would often take pictures of similar objects but have vastly different reasons for taking the photo. She also found it vital to allow participants to erase photos.

Summary In summary, feedback studies using a medium more convenient for participants, such as cell phones, may yield higher use rates. Also, recent use of photo-elicitation has shown it to be a promising method of gaining more detail about participant's everyday events, augmenting participant focus on the interview itself and encouraging participants to make new associations. This work has also shown the importance of peripheral information in photographs as well as identifying the need for participant review of photos prior to conducting the interview.

Cultural probes

In the cultural probe method researchers design a set of tools that participants use to express their feelings, beliefs, and attitudes. Considerable planning goes into the design of the probes themselves to make use of familiar functionality (such as a camera) while encouraging participants to examine their daily life in a way that have not before (such as asking them to take a photo "at 8pm on a Sunday") [67]. Cultural probes are similar to diary studies in that they are highly participant controlled, but are intended to capture general attitudes and social trends rather than everyday interactions. In a related technique, Hutchinson *et al.* introduced technology probes, or simple tools designed to encourage creative use, to a field situation to generate design requirements for more specific tools [79]. However, technology probes are not designed to capture everyday interactions.

Note that the breakdown in Table 2.1 does not distinguish between cultural probes and elicitation diary studies. In reality, each category is a scale between two extremes, and these two methods can be seen as being different with regards to capture control (experimenters usually provide more structure for cultural probes).

Other approaches

In contextual inquiries and participant observations, researchers are able to discuss the implications of events and actions with participants immediately. These studies yield the most objective data but are subject to presentation effects (*e.g., participants may act differently because of the presence of the researcher* and are time-consuming and difficult to scale. Also, a general problem with *in situ* studies is that the participant may feel annoyed if they are too distracted by questioning.

Contextual inquiry could be modified so that the communication between participant and experimenter occurs more-or-less *ex situ*. In this case, the discussion of captured events could occur significantly later than the empirical event of interest. For example, a researcher conducting a contextual inquiry may not be able to interrupt the participant because he or she is conducting a sensitive operation. In these cases, researchers may record questions, and recording tools would probably aid participant recall of the event.

Other studies could use synchronous, *in situ* communication, with remote presence, such as phone interviews held immediately after and in the context of significant events. These types of studies are highly disruptive to participants and are therefore difficult to conduct. However, if events of interest occur infrequently, this method can provide more detail than *ex situ* methods.

2.4 Challenges of prototyping

While observational techniques can help to inspire ideas and provide requirements for design, to arrive at usable interface designs, product designers commonly build a series of prototypes: approximations of a product along some dimensions of interest. Prototyping is the pivotal activity that structures innovation, collaboration, and creativity in the most successful design studios [91]. Prototypes play important roles for four distinct constituencies. First, designers create prototypes for their own benefit; visually and physically representing ideas externalizes cognition and provides the designer with backtalk [159] – surprising, unexpected discoveries that uncover problems or generate suggestions for new designs. Second, prototypes provide a locus of communication for the entire design team – through prototypes, the tacit knowledge of individuals is rendered visible to the team. Third, prototypes are integral to user-centric development by providing artifacts that can be used for user feedback and usability testing. Fourth, prototypes are also important sales tools in client relationships – many product designers live by the principle, "never enter a client meeting without a prototype in hand." Through much of the design process, designers today create two separate sets of prototypes: *looks-like* prototypes that simulate "the concrete sensory experience of using an artifact" and show only the form of a device [76]:3, and *works-like* prototypes that use a computer display to demonstrate functionality and more closely simulate actual user experience [21]. The time and expertise requirements for creating comprehensive prototypes that tie form and function together prohibit their use until late in development. At that time, monetary constraints and resource commitments prohibit fundamental design changes [171].

2.4.1 Lightweight prototypes

By lightweight prototyping, we mean the rapid iterative process of designing and exploring representations that look like or work like a possible application. Examples include sketches, paper prototype mock-ups [150, 167], probes, and Wizard of Oz simulations of working systems. All of the challenges are problematic at this stage of development. While similar challenges might exist in other domains, ubicomp developers face major development hurdles at this stage. As a result, this often becomes a bottleneck for ubicomp developers.

During the early stages of design, it is important that users do not focus only on surface usability issues such as color and typography. Thus, it is important to design lightweight prototypes that do not appear to be finished products [104]. However, it can be time consuming even to simulate core interactional features of a ubicomp system with lightweight prototypes. For example, in evaluations of mobile applications it is difficult for person to shadow users while they move, or to distribute sensed information to different sites, users, and devices.

"Looks-like" techniques that require no coding, such as graphical mock-ups, are limited in terms of realism. However, when high levels of interactivity are not necessary, they can function as informative, unobtrusive situated probes to provide realistic data on potential use. In non-situated settings, they can also provide straightforward ways to explore the impact of ambiguity (a developer could roll a dice to simulate recognition errors). "Works-like" techniques such as technology probes, if deployable, can provide situated, real information. Depending on the level of functionality, they may also be able to address ambiguity. If they function smoothly, and do not have too rough an interface, they may be unobtrusive. Prototypes that are robust enough to be deployed longitudinally are best for addressing issues of data sparsity.

Functionality of both "looks-like" and "works-like" prototypes can be enhanced with the help of the Wizard of Oz approach. Wizard of Oz was originally adopted for speech user interfaces because having a human "recognize" the speech obviates the overhead of implementing or configuring a functioning speech recognizer [39, 90, 123]. Recently, Wizard of Oz has emerged as a particularly successful technique for ubicomp, because of the number of sensors involved and the amount of technology integration often required. Early in the design process, having a wizard perform some aspect of this manually can help developers to gather user feedback quickly. In ubiquitous computing, Wizard of Oz control has shown to be useful for simulating recognizers [6])multi-modal interfaces [29, 141] sensing [77, 35, 130], intelligent user interfaces [39], location [16, 107], augmented reality [110], and input technologies [100] early in the design process. Once software is developed, Wizard of Oz-enabled tools can assist in the collection and analysis of usability data [102] and in reproducing scenarios during development and debugging [102]. Looking forward, we believe there are many opportunities for richer integration of Wizard of Oz into design tools, and for increased adoption of the design, test, analysis philosophy utilized in SUEDE [102], a tool that allows designers to prototype prompt/response speech interfaces.

Another approach to achieving realism with "works-like" prototypes is to create robust prototypes with very simple functionality that can be rapidly created and deployed to probe use patterns. The original culture probes introduced by Gaver [54] have been expanded to include technology [79, 145, 146]. Such probes can help to "achieve three interdisciplinary goals: the social science goal of understanding the needs and desires of users in a real-world setting, the engineering goal of field-testing the technology, and the design goal of inspiring users and researchers to think about new technologies" [79]:17. These technologies can gather information about sparse data if they are sufficiently robust by going beyond short deployments. Over the course of a longer deployment they will also slowly be integrated into daily life, becoming less and less obtrusive. Alternatively, a probe might be entirely simulated, as with *paratypes*.

In deciding among these techniques (paper prototypes, interactive prototypes, Wizard of Oz prototypes, and probes), a designer must make trade-offs between realism, unobtrusiveness, data sparsity, ambiguity, and cost/time. Paper prototypes and Wizard of Oz prototypes can be used to explore ambiguity (by manually or virtually "rolling the dice," respectively). Probes or other technologies that can be deployed in real-world situations over time can support both realism and sparsity. Paper prototypes and interactive prototypes may be the least costly techniques, but they may also be least flexible in addressing challenges.

Researchers have recently begun comparing the combined cost of creating and evaluating paper and interactive prototypes. In evaluating a system for locating items in an industrial-sized kitchen, Liu and Khooshabeh [108] compared paper prototyping to an interactive system that looked more finished and included some functionality. They found that more people were needed to run the paper prototype study, and that it was hard to make sure that it was present and interactive at appropriate times. However, the paper prototype took the authors only a day to create while the interactive prototype took two weeks. In a different study, Mankoff and Schilit [116] deployed paper prototypes of an application in 16 separate locations for a month. Wizards responded to user interactions once per day. The prototypes supported situated activities such as group conversations and requests for missing supplies. The time to build the prototypes and run the evaluation was minimal. One reason this worked was that the application did not require real-time responses. These examples illustrate that, if used judiciously, paper prototypes can be an effective, time-efficient method for eliciting user feedback. However, the examples show, because human labor is required to achieve "interactivity," the cost/benefit ratio is only attractive when human involvement is limited.

2.4.2 Functional prototypes

> "Effective evaluation, in which users are observed interacting with the system in routine ways, requires a realistic deployment into the environment of expected use" [5]:49.

Eventually, it becomes necessary to deploy a real prototype in the field. These prototypes go beyond the lightweight representations mentioned above to include real interaction. While high-fidelity implementation of ubiquitous computing systems deserves a longer discussion than space affords, we highlight a few particularly salient issues here: It is difficult to develop systems robust enough for realistic situations and to coevolve with user needs quickly enough to sustain a critical mass of users.

Reasons for lack of iteration include the expertise and the time necessary to build working ubicomp systems that work at the level needed by most applicable existing evaluation techniques. The process of building prototypes for realistic use can require considerable technical expertise in many different areas. One developer we interviewed commented, "I would say the hardest part about implementing these displays is the mechanics of doing it. . . ." Similarly, Hartmann *et al.* [65] found that while design consultancies have many design generalists, they do not have enough programmers and electrical engineers to complete large prototyping projects.

For a large majority of ubicomp applications, tremendous resources, expertise and time must be committed to create prototypes that function consistently across different devices and places [2]. Tools that simplify interface iteration, reduce coding, support remote administration and diagnosis, and reduce the burden of reinstallations can help. The first two solutions are important in any prototyping system. Remote administration and remote installations are particularly important to support in ubicomp applications being field tested. Researchers and developers have created some tools and toolkits to allow developers to rapidly prototype ubicomp devices for early-stage testing (including [56, 100, 117, 133, 106]). However, our interviews revealed that some developers are not taking advantage of the abstractions these toolkits provide, instead choosing to build systems from the ground up. This suggests that more work needs to be done to convey the benefits of these systems to developers and that toolkit developers may need to design more flexible systems.

2.5 Challenges evaluating prototypes in the lab

Controlled evaluations comprise laboratory experiments, field simulations, and controlled field experiments [126]. They are typically used when precision is important (*e.g.*, determining how long users take to complete constrained tasks), but are used less often to determine realistic use. Methods that emphasize realism, such as field experiments, are untenable for some applications, such as those that augment spaces for which there is an extremely high cost for any obtrusive deployment (*e.g.*, hospital emergency rooms or airplane cockpits), or that are extraordinarily difficult to simulate (*e.g.*, city transit systems). In these cases, it is necessary to address ecological validity in more controlled evaluation environments, such as labs.

Practically speaking, controlled evaluations can be very effective at testing issues of aesthetics and standard graphical interface interaction, as well as for comparing possible solutions. Running a study of this type is no different for ubicomp than for any other domain. Ubicomp developers must simply realize that they must select aspects of their system that are amenable to this sort of testing. For example, our mobile designers found controlled studies especially important when testing the readability of information on small mobile screens.

Recent work suggests that recreating the context of use through scenarios in lab settings may provide just as much or more feedback on usability problems as field experiments for some ubicomp applications. Kjeldskov *et al.* [96] found that a laboratory test approximating field use found usability problems at a lower cost than field experiments. Kjeldskov and Stage [97] also investigated more general methods of simulating realistic mobile situations. Specifically, they devised a lab evaluation approach using treadmills that involves different types of body motion (none, constant, and varying) and different attentional demands (none and conscious). Simulating these fundamental properties of the situations in which ubicomp applications are likely to be used to help extend the usefulness of controlled evaluations for ubicomp developers.

2.6 Challenges evaluating prototypes in the field

When ubicomp applications are deployed and used (or even commercialized), it gives the field valuable data about what really works or does not work. As noted above, creating prototypes robust enough for field deployment is challenging. But other challenges also make field experiments difficult, such as issues related to critical mass including adoption and extended use, data sparsity, and generalizable comparisons of different prototypes.

Critical mass is difficult to maintain in field experiments because people may be slow to adopt a technology or may be quick to abandon a technology after a small number of breakdowns. One way of addressing these issues is by making use of *local informants/champions*, people who are well-known and respected in the deployment site who can help to speed up acceptance and to increase the chances of success [27]. Another approach to addressing critical mass is the living laboratory, a later stage technique that seeks to test and iterate on ubicomp systems in an everyday context that is highly accessible to the developer/experimenter. EClass included multiple projected displays for the instructor, a large-screen, rear-projection whiteboard, pen tablets for students, video and audio recordings, and web based access to recorded data at a later time [1, 3]. It was deployed and iterated on over the course of several years in a classroom in which the developers taught and studied, as well as in the classes of colleagues of the developers. Intille *et al.* [84, 82] are continuing this tradition with PlaceLab, a living laboratory designed to sense and augment everyday domestic activities.

Events of interest may occur only sporadically or may be difficult to sense in field settings, leading to sparse data collection. One way of addressing this concern is to collect, unobtrusively, logs of all important events. For some applications, *in situ* observation can be unobtrusive, such as systems deployed in public spaces [31]. But this approach is more difficult for other types of applications, for example mobile prototypes. Methods for handling these cases include integrating data collection into the prototype [149], or adapting the needfinding techniques discussed above to encourage users to introspect on their situated use of deployed technologies.

Given that it is difficult to evaluate only one prototype, it is clearly also challenging to conduct an experiment comparing multiple prospective designs. To address this issue, Trevor *et al.* [170] developed a comparative study methodology similar to a laboratory experiment. They used quantitative and qualitative data to compare and contrast two types of interfaces: portable (*i.e.*, mobile) versus embedded. The difficulties of evaluating ubicomp applications in the field made it difficult for them to conduct a true controlled study. However, their interfaces were *designed for evaluation* rather than for use, and this allowed them to gather information that could be used for comparison. Trevor *et al.* [170]:66 gathered data about issues including usability (which they defined as "learnability, efficiency, memorability, error handling, and user satisfaction"), and utility, or "the functionality that users perceived to be useful." They also gathered data about availability, trust, and privacy, issues that may affect end-user satisfaction in ubiquitous computing environments but are not normally tested in traditional GUI applications. The deployment continued for several months, and they found a set of tradeoffs between performances on different metrics and type of interface.

2.7 Summary

In this chapter, we derived a set of challenges of evaluating ubicomp applications from a literature review and a set of interviews with developers. We also showed that developers have a small but growing set of tools to overcome these challenges: self-report methods for needfinding; Wizard of Oz, paper prototyping, and probes for lightweight prototyping; research and professional toolkits for functional prototyping; methods of recreating environments for controlled evaluations; and a set of approaches to encourage use, gather data, and compare deployments in field experiments.

Chapter 3

Extending the diary study method

In this chapter, we describe extensions to the diary study technique to better support ubicomp experimentation, derived from three studies of the technique itself in action.[1] For two of these studies we played the role of a participant observer by involving ourselves in an ongoing study. Specifically, we observed the process of using the method, analyzed results from the study and interviewed the researchers involved about their experiences. The other study we ran ourselves to gain first-hand insight into the issues involved in running a diary study and to compare and contrast the use of different capture media: photos, audio clips and tangible (physical) objects. While photo diary studies are gaining in popularity, use of the other two media is limited.

Our studies revealed a need for situated annotation of captured events in elicitation studies. We found that the best approach to feedback studies may be to combine media capture with structured, question-and-answer based annotations. Our studies also revealed the usefulness of different media in different situations. Specifically, we found that images lead to more specific recall than any other medium, but that audio, in addition to making it easier for participants to capture information that does not have a visual representation, can be used clandestinely in situations in which participants do not feel comfortable using a photo to capture an event. We found that information about location does not significantly impact recall, and that tangible objects are more likely than other media to prompt discussion of broad attitudes

[1] This chapter is based on [26].

and beliefs (Figure 3.1). We also noticed unforeseen issues in elicitation interviews. For example, while media capture lent itself to a sequential review of data, interview discussion tended to follow themes, causing problems for participants and researchers when they referenced captured data out-of-sequence.

We also developed a technique and a tool to support media-based diary studies. Our experience with media-based diary studies as well as reports in the literature, indicate that it is important to mitigate the impact of a study on participant's everyday interactions and encourage participant recall of ambiguous data. We also found it important to provide support for interview preparation. To address these issues we propose a diary study pipeline that borrows from both feedback and elicitation methods to maximize participant recall and interview preparation while minimizing situated logging. We then built and tested a lightweight tool, *Reporter*, to support this pipeline. Results showed that participants were able to learn the tool rapidly.

3.1 Studies

None of the studies cited above concentrate on how using different media for communication and prompting might affect the diary study method as a whole, including the types of responses different media elicit as well as how different media effect the process from an experimenter's perspective. To thoroughly explore media use in diary studies we analyzed the results of three studies to better understand how to support media elicitation and diary studies (see Table 3.1). The studies included an elicitation study in which participants used photos as prompts; a hybrid phone feedback and elicitation study; and an elicitation study comparing photos, audio and tangible objects as prompts. For the first two of these studies we played the role of a participant observer, observing and analyzing studies run by other experimenters. We ran the third study ourselves.

Figure 3.1. A selection of tangible objects collected by participants in the festival study. "The flowers (upper left) ... mirror how I think about jazz."

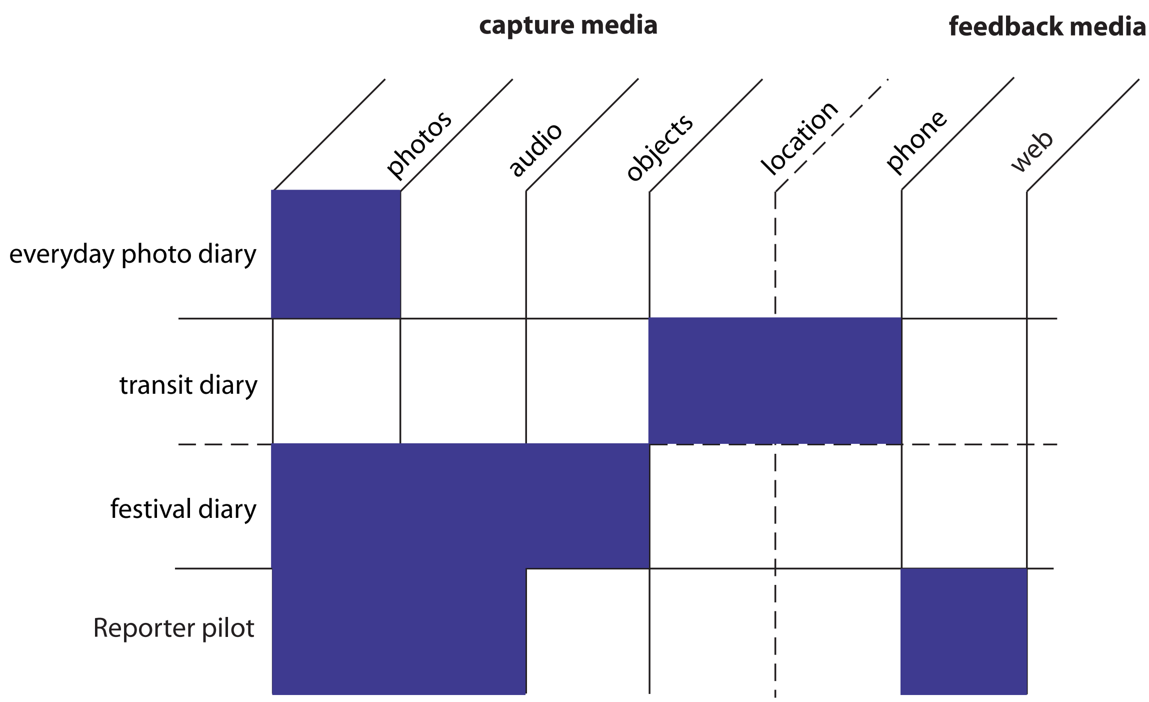


Table 3.1. We analyzed the results of three studies that used different approaches to media capture and feedback We also ran studies exploring web-based feedback methods (bottom row). In this table, the dotted horizontal line separates studies we ran (below) from those other experimenters ran (above). Also, the vertical dotted line separates capture media (left) from feedback media (right).

3.1.1 Photo diary in an everyday setting

This study was run by another group at our institution, the members of which we refer to as *experimenters* below. We were involved in the study from beginning to end, observed the process of using the method, analyzed results from the study and interviewed the experimenters involved about their experience. In the study, experimenters explored how people search for, consume and produce information. The experimenters' *participants* in the study captured information consumption or production events using digital cameras and used photo-elicitation to explore the meaning of captured photos.

Method

After piloting the study with eight participants, the experimenters recruited an additional 11 participants for this study from within the social networks of the experimenters. The participant group was split evenly between men and women. Ninety percent of participants reported some college education and sixty percent some graduate education.

The experimenters gave each participant some basic instructions and a digital camera. The digital camera (Logitech's Pocket Digital 130) was chosen as it is small, robust and easy-to-use. The experimenters asked participants to take a photo each time they noticed that they were consuming or producing information. An instruction sheet noted that things like reading a newspaper, surfing the web, reading email, watching television, listening to the radio and other similar activities were of interest and should be photographed. Also, the experimenters asked pilot participants to create written annotations for each photo on a small notepad. However, as all of the pilot participants abandoned written annotation after only a few attempts, the experimenters decided not to ask for such data of the study participants.

Participants captured photos for an entire day from the point they awoke until they retired in the evening. Because of the amount of time required for elicitation interviews, the experimenters felt it important to discourage casual picture-taking. One day after the day chosen to diary, the participant was interviewed for approximately one hour. The participant first completed a small demographic survey that included questions about their information use. The interview was qualitative in nature and revolved around the images captured by the participant.

After completing the interview the experimenters coded and analyzed both the interview and the captured photos. We also analyzed the data to construct a coding scheme focused on the method rather than the goal of the experimenter's research. We iterated this scheme a number of times with the experimenters. This iteration was done qualitatively, similar to [93]. We then classified the data according to this scheme (List 3.1.1), and asked the experimenters to classify a subset (10%) of the data. We initially found 76% inter-assessor agreement, but after one iteration increased the accuracy to 89%.

Below we present results both from our coding of the participant data as well as interviews with experimenters about the method.

Results from participants

In summary, the median participant captured 34 photos (minimum participant: 15, maximum participant: 90). In the interviews, the experimenter would progress through the photos in the order they were taken by the participant. Our analysis of the interviews and photos revealed that more time was spent with photographs viewed at the beginning of the interview than those viewed toward the end and that often participants and experimenters would reference photos out-of-sequence. Also, participants would often photograph an object that itself was meaningless but that would cue recall of a specific event ("pointers"). We also found several cases in which participants adjusted their photo-taking process because of the presence of other people. Other findings included a prevalence of dynamic objects that do not lend themselves to photos, peripheral information in photos that was important in the elicitation interview and many instances of staged photos (*i.e.*, photos in which participants arranged objects or people specifically to make them easier to photograph — see Figure 3.2).

We found that in every case, interviews followed themes rather than the temporal order of capture. Thus, the experimenters reported that "most [of the interview] was

Listing 3.1. Codes for the photo elicitation study

```
recognition
recall about a specific object
recall about the surrounding context
verbally described sequences
photos of sequences
absence of photographable objects
absence of non-photographable objects
absence (other)
cues related to other media
attitudes towards capturing people/power issues
evolution of capture attitudes
general versus specific recalling
overloading
containers
pointers/stand-ins for information
people
objects
written communique
peripheral objects
media (as well as type)
```

Figure 3.2. For the photo diary study, participants were asked to capture information as they encountered it in everyday settings. Participants occasionally captured important information unintentionally. A participant intending to capture an Internet search also unintentionally captured paper-based reminders (top). Some photos referred to events that were not visible. A participant took a photo of a door to signify that he was having an argument with someone on the other side of it (bottom).

spent on the first one-fourth" of photos taken because the first photos would touch off a discussion about general habits that would not be revisited upon seeing later photos. For example, if someone got a cell phone call in the morning, a photo of that event would lead to a general discussion of cell phone use. In that case when photos of cell phones were viewed later in the interview, they generated far less discussion. Also, for some participants the interview would begin to concentrate on the topic of a later photo. In these cases, the experimenter or participant would often "save that topic for later" and continue with the discussion with the intent of picking up that thread of conversation again when they arrived at the appropriate photo. Six out of 11 participants showed this trend, with total occurrences ranging from one to six times during their respective interviews.

Nine out of 11 participants used pictures of pointers, or objects that were not themselves information events but were reminders of other events, from three to 10 times during their interviews. These objects that served as pointers were usually related in a peripheral way to the event itself. For example, in one case a discussion with a colleague about an article read online was recalled via a picture of the beverage the colleague brought for the participant at the time. Also, there were five instances of pointers that were themselves direct side effects of the information event captured, such as a photo of diagrams on the wall as a reminder of the information conveyed during the meeting that produced them. Each of the three times a pointer referred to a daily event (*e.g.*, getting the morning paper) the pointer led to only general recall of the event, with participants using such qualifiers as "usually" without uncovering the specifics of the instance. In every other case in which pointers were used they referred to unusual events and recall was specific.

In addition, we found that six out of 11 participants adjusted the style of capture because of the presence of people in the picture from two to seven times per interview. In one such case, a participant took a picture clandestinely to avoid further aggravating an angry family member. Also, during the interviews the experimenters conducted, there were two incidents of tangible objects serving important but unexpected roles as prompts. In one case in particular, the poor physical appearance of a participant's cell phone prompted follow-up questions about the participant's attitude toward the device.

Similar to others who have used the photo-elicitation method, we found that the most important information gleaned from the interview usually came not from the photos themselves but from the participant's description of the actions and processes that led them to take that image. However, we did note in seven different cases participants were unable to recall why they took an image. This was usually due to either the image resolving poorly or the capture of overloaded objects. For example, one participant took several pictures in a row of his cell phone. The elicitation session revealed that he had answered several phone calls in a row, but could not remember the specific content of those calls.

We also recorded the following findings:

- In all participant interviews there were situations in which peripheral information cued important discussions (*e.g.*, something not intentionally captured became a topic for discussion). The number of references to peripheral information ranged from one to 11 times per interview with a median of three occurrences. In nearly all of these cases, the experimenter, not the participant, first referred to the peripheral information during the elicitation interview.

- In all participant interviews there were instances of staged photos, or those in which the participant arranged the scene or in which a person was photographed presenting for the camera. The number of staged photos ranged from two to 26 times per interview with a median of nine occurrences.

- Eight participants referenced other media in their photos. These were events in which participants took a picture of a physical artifact of some other medium, usually audio. Of the participants that showed this trend, the number of referenced media ranged from one to four times per interview with a median of 1.5 occurrences.

- Ten participants took photos that did not record correctly, usually because of lighting issues. Of the participants that showed this trend, the number of improper recordings ranged from one to five with a median of two. However, in nearly all cases the photos nonetheless led to recall of a specific event.

- Nine participants referenced some object in the interview that was itself never captured. Of the participants that showed this trend, the number of non-captured references ranged from one to nine times per interview with a median of four occurrences.

Results from experimenters

From our interviews with the three primary experimenters running the study we found a need for situated annotation as well as a means to review captured data and annotations before the elicitation interview takes place. The experimenters reported that they spent too much time on just a few images and having the chance to review and categorize them would facilitate their getting the most data out of limited interview time. However, they noted that in many cases the thing being photographed was not necessarily evident and thus some form of annotation of the photo would be crucial for them to categorize photos appropriately. They also commented that "written responses" to the photos would be helpful as well, but that it was unlikely that participants would complete such questions in the field. To that end, they expressed interest in a software tool to support desituated photo feedback, but were concerned about limiting their study population to people who have access to a computer. To remedy this problem, one experimenter suggested a "computer that is publicly available that has this kind of photo feedback software" that participants could easily access.

3.1.2 Transit decisions diary study

This study was run by a colleague at our institution. Similar to the first study, we were involved in the study from beginning to end, observed the process of using the method, analyzed results from the study and interviewed the experimenter about his experience. This was a hybrid feedback and elicitation study that explored how people make public transit decisions. The experimenter used phone-based feedback as well as location capture for elicitation. We analyzed feedback from participants as well as the results of the elicitation interview. We also interviewed the experimenter about his experience with the method.

Method

The experimenter provided four college students with a cell phone for a two week period and asked them to call a specified number every time they made a transit decision. When they called they were led through a series of questions about the event. Also, the location from which they placed the call was automatically derived from a built-in GPS sensor and communicated to our server. The experimenter then conducted two elicitation interviews: one a week into the study and the other immediately after the study was complete. During the first interview, the experimenter used transcriptions of participant recordings as prompts, and during the second interview he used both the transcriptions as well as maps indicating the location of the participant when he or she completed a response.

Participants were asked the following questions each time they called: 1) Where are you going to and coming from? 2) How are you traveling? 3) What are you doing during your travel? 4) Do you expect to arrive early, on time or late? 5) How long do you expect to wait? 6) Did you consult any resources when you were planning this trip? 7) Is there anything special about this trip?

Results

As mentioned, feedback studies tend to place a heavy burden on participants because they require participants to switch tasks at the moment that a particular type of event occurs. However, the events about which participants provided feedback, transit decisions, occur relatively infrequently, reducing the burden on participants and yielding relatively high response rates.

Two participants logged responses a median of four times per day (Monday-Friday) while the other two responded a median of two times per day. The responses generally occurred at the beginning and end of the day, corresponding to morning and evening commutes. However, in some of these cases participants responded only to correct a perceived mistake in an earlier response, and removing these repeat responses moves the median responses from four to three per day for one participant. Weekend response rates and times were much more sporadic, ranging from zero to two per day for one participant to one to five for another and with no specific pattern for any participant. The median time for individual participants to complete a set of answers per call ranged from one minute 37 seconds for one participant to two minutes one second for another.

An interview with the experimenter revealed that, while referring to transcribed responses was helpful in recreating specific recording events during elicitation interviews, the maps neither aided participant recall nor were helpful to the experimenter for logging purposes. Several reasons were given for this, including that "location was not reliable," and therefore not always captured for every event because of GPS coverage issues, that the maps could not dynamically show a sequence of calls and that the maps lacked detail. The experimenter noted that, "being able to visualize the sequence of responses would have ... made it ... simpler to reconstruct the event," compare it to other possible events, and ask the participant about their choices.

3.1.3 Festival diary study

To gain first-hand insight into the issues involved in running a diary study, we ran a study ourselves based on our first two studies and using the diary study method. During the study, we took notes on methodological breakdowns. From a sociological perspective, the study was designed to understand people's experience of novel information in non-everyday settings. From a methodological perspective, we compared the standard medium for elicitation studies, photography, with two other media, tangible objects and audio. As a contrast to the previous two studies, we chose to look at an unusual situation, a festival, rather than an everyday context.

Method

The focus of this study was to understand peoples' experiences of novel information in non-everyday settings. We ran the study in a different context and with different media capture devices from the previous studies. We recruited seven college-educated participants, five women and two men, to capture information-related events they experienced during one day at a nearby jazz festival. We divided the partici-

Listing 3.2. Codes for tangible objects

```
information
pointer
side effects
natural
man–made
general
attitudes
specific event
arranged
recall
recognition
```

pants into groups: two of the participants used digital cameras, two used digital audio recorders (Aigo mp3 player/recorders) and two were asked to collect tangible objects in a bag. Also, we asked one participant to capture both audio data and tangible objects.

After we provided tutorials on the use of the capture technology, participants captured information during one day of the festival and were interviewed about their items immediately afterwards. After completing the interviews we coded the interviews and the captured media and we cataloged all of the objects collected by tangible-media participants. In this case we coded the data only for issues related to method, especially those that had to do with differences and similarities in capture media. Our coding scheme for images followed the one we developed for the photo elicitation study, and we developed a new schemes for tangible objects (List 3.1.3) and audio (List 3.1.3). We recoded the data after a week interval and found a intra-assessor accuracy of 96%.

Figure 3.3. Tangible objects collected by participants in the festival study. A participant wore the flower (bottom) as a way to “show [friends] that I am not always prim and proper.”

Listing 3.3. Codes for audio events

```
music
ambient
conversation
unknown event
clandestine
annotation
recall
recognition
```

Results

During the one day of the study, the photo-elicitation participants collected 56 and 42 images, the audio participants collected 25 (median length: 1 minute and 32 seconds) and 45 (median length: 21 seconds) recordings, and the tangible object participants collected 28 and 14 distinct objects. Also, the seventh participant collected 12 recordings (median length: 1 minute and 3 seconds) and 13 objects.

Tangible objects Of the 55 total tangible objects the three participants collected (see Figures 3.1 and 3.3, 30 were information objects themselves (*e.g.*, flyers), 14 were pointers to some information event while 11 were side effects of some event, for example an extra copy of a form that a participant completed (see Appendix D). Also, because this was an outdoor event, many of the objects collected were natural objects, such as a leaf from a tree, but represented entirely different events and media (the leaf was used to both represent music as well as an encounter with a friend). Also, the type of events prompted by the tangible objects varied considerably. In one case, a piece of bark prompted a participant to discuss in considerable detail a complicated event in which she helped a handicapped festival-goer physically maneuver in a crowded spot (see Figure 3.1, upper left). In another case, a participant grabbed three flowers to represent a general idea she had written in a personal diary during the festival.

Also, during the elicitation interview we found the participant's spatial arrangement of the items to be important. In the case mentioned above, the participant arranged the flowers on the table in a gradient from brightest to darkest and explained that the arrangement mapped her opinions of various flavors of jazz: Latin (bright), cool jazz (middle) and traditional (dark) (see Figure 3.1, upper left). Also, another participant arranged all of her items during the interview by narrative. For example, one of her narratives involved being asked to dance by a man, and she used both a bottle the man had with him and a jalapeño he gave to her as prompts during her explanation of the story (see Figure 3.1, upper right).

We noticed that memory of event order was poor. Two of the participants corrected their recall of the order of events three times and the other made four corrections. Also, two of the participants were not able to recall any details about the ambient audio at the time of capture, and the other participant recalled ambient audio roughly half of the time.

Audio recordings The type of audio that was recorded varied significantly by participant. The first participant captured almost entirely (23 out of 25) music events, while the second captured almost entirely (40 out of 45) ambient events, such as people talking, planes going by or overheard conversations. The second participant tended to capture more music events (8 out of 12).

We found that after identifying the contents of the recording, participants had excellent recall of the event. During the interview we stepped through each of the recordings in the order that they were made, so all participants were able to recall the sequencing of events. Participants universally recalled the place the recording occurred. However, participant *identification* of the event took longer than anticipated (45 seconds of playing time on average). While recording quality was often the reason for poor recognition, in some cases the event of interest to the participant simply did not occur at the beginning of the recording. In those cases, participants tended to describe general features of the event ("This sounds like a bunch of people talking, and I was really interested in their voices") until some point in the recording when they were suddenly able to recall the specific event and then describe that event ("Oh, laughing, that's it, I was fascinated by how these two people were laughing...").

We found that participants used audio to clandestinely capture events that they otherwise may not have. One participant captured 12 events of other people talking, masking her recording by pretending to be doing something else, for example hiding the recording device behind a book or in her palm while looking in another direction. In these cases we asked participants how they would have captured such data if they had a video device. In most cases the participants said they would not have recorded the event unless they asked permission first, in which case, as one participant said, "it would not have been very realistic."

Also, during elicitation, the participant who used both audio and tangible capture referred to some events being "linked," or audio that annotated a tangible object, and we spent some time searching for the appropriate audio recording before finding it.

Images Participants captured 43 and 55 images. Unlike the everyday study, the interviews largely followed temporal order of capture more than the thematic order of capture, likely because most images were of one-time events. Also, there were only five total pointer images. That is, most of the photos contained information about the actual event that the participant experienced.

There were 37 total media cue events, but this was likely due to the fact that the music had a physical representation (the performers themselves). However, audio recall was poor when cues were not provided in the photos themselves.

Though participants rarely adjusted photos taken of another person out of privacy concerns, there were several other occasions in which the participant's picture-taking affected others. For example, one participant clandestinely took a picture of a man who was sketching a young woman sitting in the crowd, unbeknownst to the woman. The woman, seeing the participant take the image, began to take an interest in what the man was doing and ultimately the man offered his sketch to the woman.

Finally, peripheral information in the images was again significant, playing a role in 20 images.

Comparing results between media in the festival diary study

The results from the festival study allowed us to compare audio, photo and tangible object elicitation. As Barsalou argued, it is important to analyze to what extent each capture medium supports recognition of *who*, *where* and *what* information about the captured event [12]. In each case, participants were able to recall people involved in an event with whom they were already familiar. Photos provide the best support of *who* and *where* recognition, while audio clips, once recognized, also provide adequate support. Tangible objects did not lend themselves to *who* or *where* recognition.

Timing and sequencing of events are important for activity reconstruction. Participants tended to have poor recall of the exact time of capture for all media. Also, for tangible objects participants were not able to recall correctly the complete sequencing of events, while photo and audio capture are inherently sequenced.

In addition, participants generally were unable to recall information on media channels other than the cue. When a visual representation of the media was captured, it was often difficult or impossible to recognize and participants often could not recall the specific media. For example, a participant in the everyday photo diary study who took a picture of his audio-playing software could neither recognize (because of the fidelity of the picture) nor recall what music he was listening to. However, the photo did lead to a general discussion about audio consumption habits.

3.2 Discussion

Results from the use of different media in diary studies suggest adjustments to the method to better accommodate different situations. Specifically, we found that audio elicitation suffers from recognition problems but encourages more clandestine capture events. Also, tangible objects are more likely than other media to elicit from participants creative explanations of attitudes and beliefs. The results also revealed the need for new tools to support the method. In particular, we found that tools are needed to support tagging of tangible objects; lightweight, situated annotation;

experimenter review of captured events; and automatic time stamping for all captured events.

Our results suggest that, overall, photos are the easiest to capture and recognize. However, audio cues can allow participants to capture events clandestinely that they otherwise may not have. Also, audio is a lightweight media appropriate for annotation. In general, for studies in which detail is important, a hybrid photo/audio capture medium is most appropriate.

From the transit study, we found that raw location information is not likely to lead to better recall of an episode. However, we also found that participants seem to be willing to spend longer answering feedback questions when the rate of events to report is low. These two issues suggest that situated feedback may be appropriate for some studies, but that feedback should be tied to better prompting cues. One way of supporting this would be to use photo-elicitation combined with more structured annotations, in which participants are encouraged to answer a set of specific questions.

Similar to location information, tangible objects are not likely to cue episodic memory. Thus, tangible objects are not appropriate for studies in which detailed recall is important. However, this lack of specificity could be a benefit for studies that concentrate less on the reconstruction of specific events and more on participant's attitudes and beliefs.

We found that similar to cultural probes, tangible object elicitation may inspire unique ways of describing and codifying beliefs and behavior [54]. Also, while cultural probes are able to inspire responses to general feelings about a community and culture, auto-driven object elicitation helps inspire recall and description of specific events. We also found that it would be useful to tag objects with audio annotations.

To support recall of ambiguous events, we found that it is important that each capture event be tagged with a brief annotation. However, as noted above, in the photo-elicitation study pilot participants who were asked to annotate their pictures with written diaries usually gave up the practice immediately as it was too disruptive. Thus, rapid, situated annotation, such as audio for photographs and tangible objects, is crucial. We also found that experimenters need to be able to review captured data as well as annotations before the elicitation takes place. This preparation is necessary to get the most out of limited interview time as possible. Because peripheral information in photographs consistently provided useful information, a tool that allows participants and experimenters to annotate various parts of the photographs would be useful. In general, the results of our studies argue for lightweight capture tools combined with lightweight *in situ* annotation as well as support for more thorough *ex situ* annotation and review by both participants and experimenters.

Our studies and others indicate that people are not good at judging how long an activity takes to complete [88]. Thus, automatic time stamping as often as possible is crucial. Our experience also revealed that in studies of everyday situations that depend on empirical evidence, rather than general participant attitudes, experimenters should only encourage the capture of "pointers" to events when those events are unusual. Capture of regularly experienced events should be direct to avoid generalization.

3.3 A proposed diary study pipeline

As we mentioned earlier, feedback diary studies and elicitation diary studies represent a tradeoff between accurate recall but burdensome logging (feedback) versus potentially inaccurate recall but unobtrusive logging (elicitation). In our experience with media-based diary studies as well as reports in the literature, we found that it is important to mitigate the impact of a study on participants' everyday interactions and encourage participant recall of ambiguous data. We also found that for elicitation studies it is important to provide support for interview preparation. To address these issues we propose a tool-supported diary study pipeline that borrows from both feedback and elicitation methods to maximize participant recall and interview preparation while minimizing situated logging. Specifically, the pipeline includes (1) lightweight *in situ* capture by participants augmented with (2) lightweight *in situ* annotation at the time of capture to encourage recall, followed by (3) more extensive annotation by participants at a later time, allowing for (4) review of the data by experimenters to better structure (5) a post-study interview (see Figure 3.4). This pipeline minimizes the extent to which participants are distracted from their primary tasks while still allowing them to recall and comment on the event at a more convenient time. Furthermore, unlike any previously conducted media-based diary study, experimenters have the opportunity to prepare for elicitation interviews based on specific data.

3.4 Tool support

As a first attempt to support the pipeline described above, we built Reporter. This tool can aid experimenters performing diary studies that use capture technologies, but it is not a replacement for other tools and methods necessary to conduct a study.

3.4.1 Reporter implementation

Reporter is a lightweight tool that combines a photo upload client, a database, and web interfaces to facilitate diary studies that involve digital capture media. The upload client is a standard Java desktop application and supports a simple drag-and-drop interface. The upload client sends photos to a remote MySQL database that associates each photo with a timestamp, UID, and the appropriate participant ID. The web interfaces are PHP-based and run on the same server as the database. The experimenter's web interface supports storing per-capture feedback questions for participants, reviewing all participant captures and comments, and annotating participant captures. The participant's interface shows captured media and associated feedback questions.

Our design also includes support for two types of photo annotation. First, users may use the Java client to upload digital audio clips captured in the field. Second, we provide a way for the experimenter to ask participants to tag parts of photos using semi-transparent rectangles that are movable and scalable. The photo annotation rectangles are implemented in DHTML and thus do not significantly impact the load-time of the page nor require any specific plug-ins. Also, clicking on the rectangles rotates through a series of colors, allowing participants to group annotations by color. As an example, consider Figure 3.5. Here a participant has used the photo annotation tool to designate features requested by the experimenter: the object that

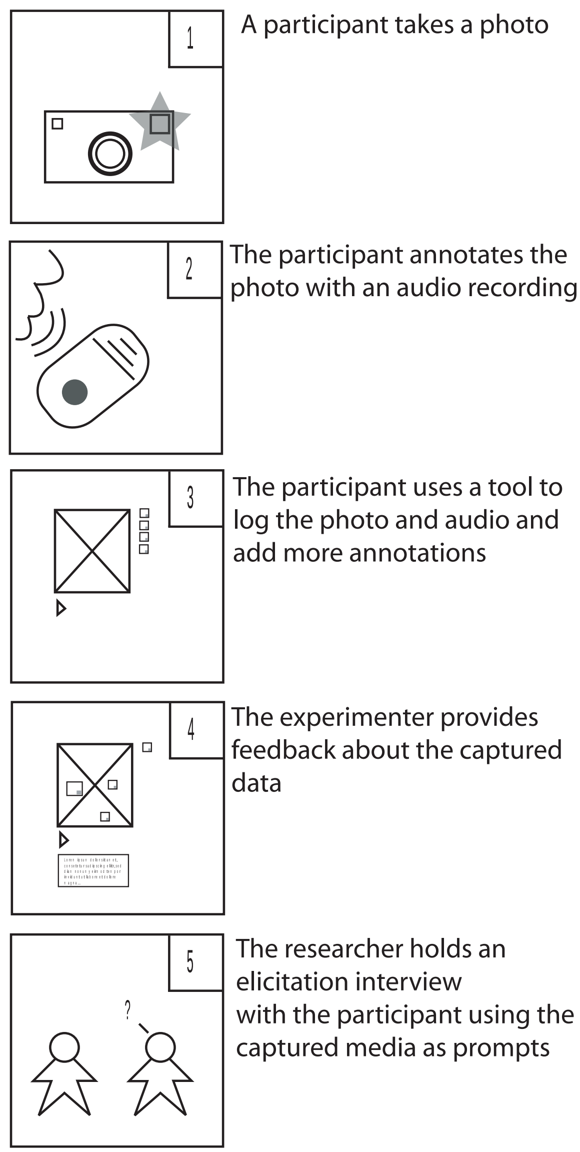


Figure 3.4. Proposed media elicitation pipeline.

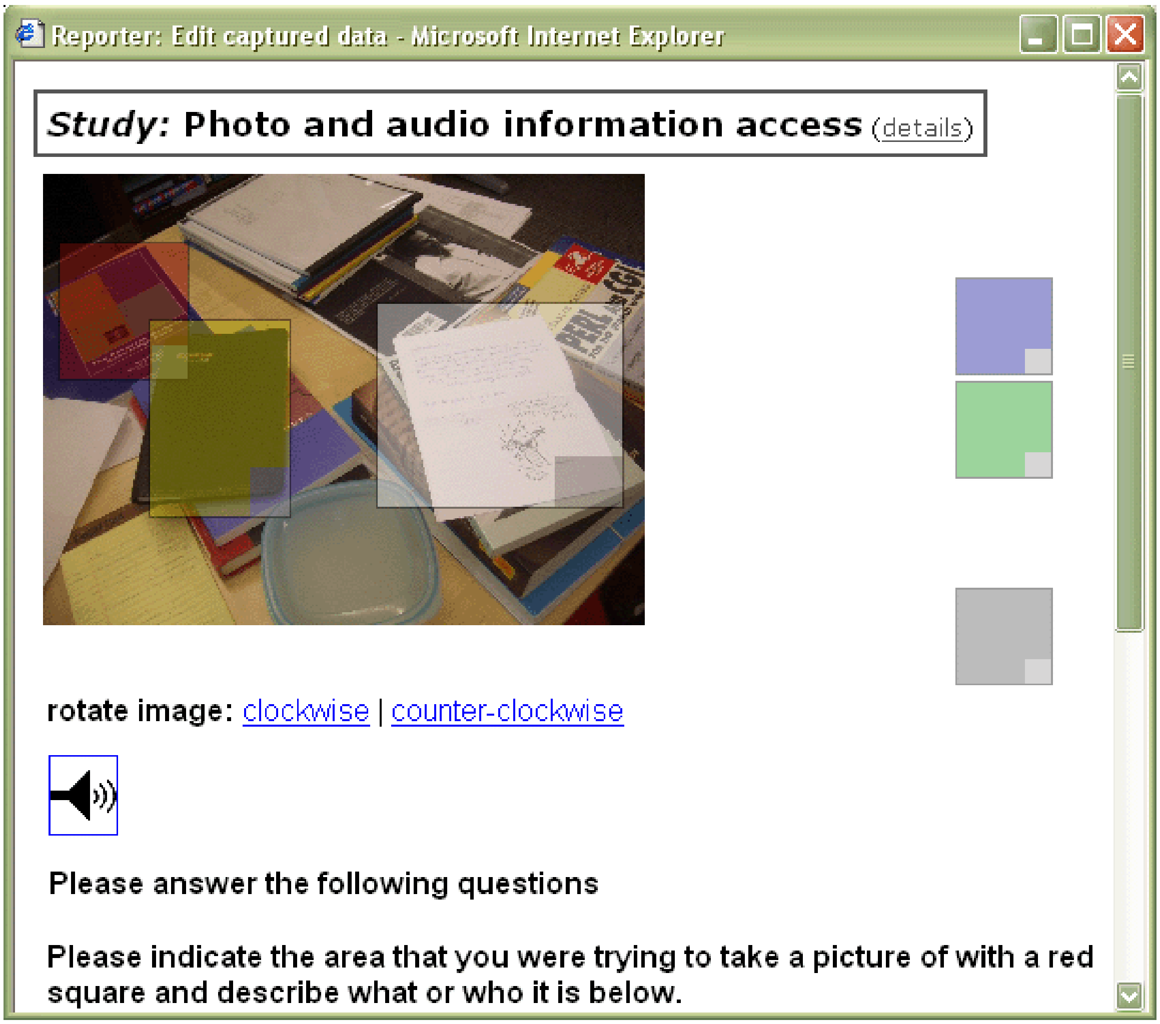


Figure 3.5. Reporter's Web interface. The participant has repositioned and resized rectangles as per questions asked by experimenters, including the object that the participant intended to capture (upper left of the photo) and any other information that the participant felt was important in the scene (center and right of the photo).

the participant actually intended to capture (center of the photo) and any other information that the participant felt was important in the scene (left of the photo). The participant has also uploaded an audio annotation captured in the field (lower left of the figure).

3.4.2 Reporter use example

The steps in a study that would use Reporter are the following: 1) A experimenter enters questions that participants will answer about each piece of captured data using Reporter's web interface; 2) Participants download and install a small Java client to a desktop machine; 3) Participants capture events and audio annotations in the field as per the experimenters instructions during some period of time; 4) When the participant is able to return to her desktop she gets data from her devices to her desktop (*e.g.*, via Bluetooth or a mounted drive – Reporter is agnostic about how this happens) and drags captured data to the Reporter desktop client and then uses a web interface launched from the client to answer per capture questions about each captured photo; 5) The experimenter uses the responses and photo data to structure a subsequent post-study interview.

If the study occurs over the course of several days, the experimenter may use the web interface to provide feedback or ask follow-up questions of participants about specific photos. After the experimenter attaches a feedback question to a photo, the web interface flags that photo so that participants can rapidly review all of the outstanding follow-up questions they have yet to answer. Also, a experimenter may use the tool completely asynchronously (*i.e.*, as a feedback tool rather than an elicitation tool).

3.4.3 Pilot test

We pilot tested the five-step pipeline and Reporter with two participants. In this study, participants captured information production and consumption events for one day and used Reporter to upload pictures and audio annotations, visually annotate photos, and answer a few questions about each event. Participants used digital cameras with audio annotation features to record events. We asked participants the following questions *post hoc*: 1) Please indicate the area that you were trying to take a picture of with a red square and describe what or who it is (in the textbox) below. 2) If there is anything in this photograph that you want to label and comment on, please use yellow colored squares and discuss them (in the textbox) below. 3) Did you talk to any people in this photograph? If so, use blue squares to designate them and indicate who they are (in the textbox) below (not names, just how they relate to you). If not, write "none." 4) Please discuss briefly how important this object/person is to you. 5) How often do use this object/person as an information resource (only once/hourly/daily/weekly/monthly/yearly)?

The day after we reviewed the data and used Reporter's feedback feature to ask follow-up questions about certain photos. The participants then answered the questions using Reporter's Web interface and we then interviewed them about the photos they captured as well as their experience using Reporter.

The participants found the system easy to learn. In particular, participants mastered the visual photo annotation technique within minutes. Also, we found as experimenters that we required a means to add links to specific study instructions at pertinent parts of the interface. For example, when uploading photos participants need to know the policy on photo deletion and modification, an issue which is likely to change per experiment. Based on this, we modified Reporter to allow experimenters to enter a study description that participants would see on every page in the web interface.

We also found that audio recording is a mostly unobtrusive means of annotating. In some cases, participants took a photo and simultaneously cued the audio recording while continuing with the task in which they were involved. For example, one participant snapped a photo while walking and carrying on a conversation with a friend. The participant then continued walking and conversing while cueing the audio recording and used a break in the conversation to record the annotation far from the point of capture. Other types of media, such as video, would likely not have been able to support this kind of use, instead requiring the participant to stop what they are doing completely in order to capture and annotate the event. On the other hand, one participant commented that he was uncomfortable recording audio annotations in some locations, such as lecture halls.

Because we were able to view photos and annotations beforehand, we more effectively structured our elicitation interviews. Though we still found it useful to understand the sequence of events that participants captured, by organizing the interviews around themes we were able to cover all of the most important types of events. In some cases, the same photograph was important for different themes. For example, a picture of a computer also included peripheral audio cues, and we discussed each at different points in the interview. Note that Reporter did not specifically support organizing photos in themes – we used pen-and-paper to compose lists of capture IDs relevant to each topic. While this sufficed, it suggests that a simple tagging scheme would be a useful addition to Reporter.

3.4.4 Field study using Reporter-based tool

Another group of experimenters used a tool based on Reporter to conduct a two-week-long diary study of childrens' approaches to learning new technologies. The tool was reengineered to better fit preexisting system components, but the only differences between this tool and Reporter were that the tool did not include a Java upload client (all media uploads were done through forms added to the web site) and the tool included support for text-based annotation but not DHTML-based layer annotation.

Four participants were recruited for the study. Two experimenters gave each participant one-click digital cameras and asked them to record every situation in which they interacted with technology. Experimenters discussed use of the cameras and upload procedure with participants, and interviewed participants twice over the course of the study. They asked participants to upload photos on a daily basis,

This study was particularly challenging because all previous studies of the tool had been with adult participants. Nonetheless, the experimenters found the tool to be useful and an easier means of gathering data than conducting lengthy participant observations of each participant.

The experimenters also found some issues that could be improved in the next iteration of the tool. For example, participants reported some confusion regarding the process of getting photos from the devices. This meant that experimenters were infrequently able to see what participants were diarying, they were unable to provide feedback to participants, and as a result much of the data fell outside of the scope of the study. For example, many participants interpreted the study as a scavenger hunt – taking pictures of items that they felt were technology – rather than taking pictures only of items they actually used. This issue could be addressed by a system that automatically collects and uploads photos (*e.g.*, using a Bluetooth connection between the camera and a computer).

3.4.5 Discussion

Our experience with Reporter focused our efforts on supporting immediate synchronization. As we present later in the thesis, we developed a system, Momento, that addresses this issue using mobile phones, which provide better support for media based diary studies: uploading is automatic, experimenters can provide feedback sooner, more information (such as GPS location) can be captured, and experimenters can potentially make use of devices participants likely to own. Momento can accept media messages from mobile devices or e-mail. The suite also includes a mobile application that can be used to attach context data to diary entries.

3.5 Summary

In this chapter, we described studies that helped us understand how to modify the diary study technique to better support ubicomp experimentation. We also prototyped some lightweight tools to support this technique. In the next chapter we combine our experiences with the diary study method with our work in Chapter 2 to derive implications for tool design.

Chapter 4

Implications for tool design

In this chapter we identify usability requirements for ubicomp experimentation, unifying findings from our interviews, diary study work, as well as an investigation of support for coevolution, which our literature review revealed is central to sustaining critical mass in field deployments. We then discuss other systems with these metrics in mind.[1]

4.1 Evaluation and design for coevolution

Increasingly, ubicomp developers are creating devices embedded in everyday practices. Such a tight link between a device and its environment suggests that in some circumstances a particular feature will become more or less important, requiring approaches to design and evaluation robust to applications whose functionality and utility is in constant flux (see Figure 4.1).

Orlikowski [139] introduced the notion of structuration of technology systems, emphasizing the duality of technology — that organizations and users influence technology and vice versa. She found that the specific context of an institution and workers within it will collectively mediate a relationship with a technology. In a deployment designed to explore these issues, she found that some managers were using

[1] This chapter is based on [28].

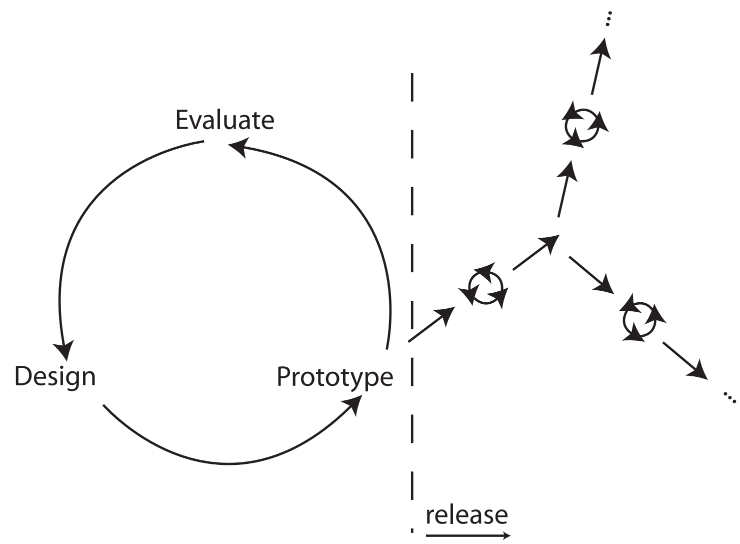


Figure 4.1. Extended iteration cycle for situated applications.

the technology as a means of social control. That is, even though workers could adjust some settings managers told them they could not, leading to "trained incapacity." In this way, institutions can embed social properties in tools not designed with this functionality in mind.

Mackay extended Orlikowski's [111] work and studied how users customize software. She found that people often appropriate software by editing preferences. In those cases, users both adapted to and appropriated the technology. Furthermore, she clarifies that appropriations may be innovations that can change the technology and thereby communication patterns. Her analysis of a two year study of the Information Lens system showed that:

External events increase chance of customization.

People change preferences when they discover patterns — "The most common ongoing customization occurs when the user becomes. aware of a commonly-repeated pattern of behavior and encodes it as a customization" [111].

A small group of users will often be the first to test new customizations. They rarely get feedback on how well these novel adaptations work, however.

Other, more mainstream, users translate individual needs to create sets of customizations organized to meet those needs.

In general, Mackay suggests "observation of use in the field as an important input to future development" as well as software support for sharing customizations. Mackay's findings and subsequent recommendations echo Lucy Suchman's [168] finding that people tend to change their categories over time.

But if users can adapt to technology, what is the real problem? Postman and Ackerman have argued that ignoring possible co-evolutionary effects may lead to systems that are inflexible or that encourage behavior contrary to intended behavior. Postman states that "I just don't think we can go into [designing distributed systems] anymore with our eyes closed" [148].

Situated activity refers to the actual means by which a particular action is done rather than any official or conveyed means. Such activities are naturally highly dependent on the specific context and content of the actions [45]. Furthermore, because "situations in which human-computer interaction takes place are increasingly varied, as computers become highly portable and embedded in everyday environments" [158] designers of novel devices must take the situation of their use into account. Situated devices, technologies whose use is greatly correlated to context, face co-evolution problems beyond the more traditional interfaces Orlikowski and Mackay studied. While many such devices use context information to adapt parameters to new situations [42], changes to the device's general functionality may require redesign.

Both tools and processes are needed to support the co-evolution of situated devices. Processes are needed to handle a design cycle that can support multiple versions of the same technology. In particular, designers may need to reconsider the standard iterative design-cycle (design, prototype, evaluate, repeat). For situated devices, this cycle may need to branch to include specific needs that arise in different situations (see Figure 4.1).

However, field studies, as mentioned earlier, are already dauntingly difficult and time consuming. Some researchers have approached this problem by integrating ubicomp systems into on-site work processes [127, 84, 1]. However, these approaches do not address evaluation support covering situations in which one is not necessarily embedded. Therefore, new tools are needed to aid field study evaluations. One possible approach is to deploy tools supporting quantitative evaluation that could feed into qualitative inquiries. For example, logging toolkits could be used to determine device use patterns while Allanson's elecrophysiological toolkit [7] and eye tracking technologies [86] could also help determine how people respond to devices in different scenarios. With such in-the-field use patterns identified a priori, ethnographers and developers would be better prepared to adapt their technology to different situations.

4.2 Requirements for participants

For participants, minimizing the burden of experimentation can help with participant compliance, adoption, and retention:

4.2.1 Unified client system

A unified client system for gathering multiple types of data (such as self reports, logged events, and ESM) is one way of increasing consistency and mitigating adoption and retention issues raised in our interviews.

4.2.2 Leverage existing devices

Our field work confirmed that carrying a new device, even loaded with useful new features, is burdensome for end users. It is important to leverage existing devices as much a possible. Communications with end users can piggyback on commonly available devices such as camera phones and devices that support text messaging.

4.2.3 Multiple, lightweight communication options

Related to this, it is important to provide multiple kinds of support for communication between participant/device and experimenter. Our diary study observations showed the need for ongoing availability of communication and asynchronous communication (*e.g.*, annotation); and both our diary studies and interviews emphasized the need for communication to be lightweight. Given a focus on participants' existing devices, options include: live phone discussions on a participant's phone (hard to manage asynchronously but allows rapid communication); text messaging (ideal for lightweight experimentation with participants who are highly comfortable with text messaging); or a custom client (requires software installation, is easily customizable by the experimenter, can include many more types of media and can be optimized for participant usability).

4.3 Requirements for experimenters

For experimenters, it is important to support different levels of computer experience, to support multiple types of data and context, and to support monitoring and notification. Some specific requirements arising from our studies include:

4.3.1 Support qualitative data, quantitative data, and contextual data

Experimenters reported valuing both qualitative and quantitative data, including diary data, ESM data, log data, and sensed context. Integrating support for multiple types of data can help to increase consistency and usability.

4.3.2 Do not require fully implemented applications

Participants in our observations wanted to test ideas they could not easily implement. To support experimentation at the early stages of design, as well as to support experimenters with limited coding experience, it is important not to require complete applications. One way to facilitate this is to support a Wizard of Oz protocol in which experimenters can do some of the work normally done by an application.

4.3.3 Support the full experimental lifecycle

The experimental lifecycle described by our interviewees included experimental set-up, modifications to an experiment, running an experiment, and analysis and summarization of data both during and after the experiment was run. A usable system should support this entire cycle.

4.3.4 Support monitoring and notification

As reported and observed in our field work, experimental data may arrive in occasional bursts, reflecting the variable nature of day to day life. When a study takes place over days or weeks, constant attention may be an inefficient, difficult way to watch for rare or uneven data. Overview displays and notifications can address this problem. For example, an experimenter might be notified when a user enters a certain space, or might monitor a display for big changes in amount of activity. These pieces of information could help an experimenter decide when to take direct action (*e.g.*, contact the user, go somewhere to make observations, *etc.*).

4.3.5 Support lengthy, remote studies

Our fieldwork identified a strong preference for remote data gathering, both to reduce observer effects and to get data of an appropriate quality. Data from studies that take place over days, weeks, or more may overcome issues such as the novelty effect and learning. Gathering situated data over time is far more feasible when experimenters can be remote. Additionally, remote experimentation helps to mitigate the effects of an observer being present.

4.3.6 Support coevolution over long-term studies

After lengthy deployments, requirements for a system might diverge. It is important that experimenters have the flexibility to morph rapidly their designs to match the needs of different environments.

Based on these requirements, we created a tool, Momento, supporting situated experimentation. Our process involved iterative studies with the tool, run by ourselves and others. After each study, we reflected on the participant and experimenter experience. The current version of Momento is described in the next chapter, while studies run during and after the implementation of Momento are described later.

4.4 Related work

Situated experimentation is an area receiving much attention in ubicomp and mobile device research. Laboratory studies can uncover some interface and navigation flaws [89, 17], especially when the physical configuration of the study embodies some aspects of a field setting [96]. However, as Zhang *et al.*'s review of emerging mobile and ubicomp research trends show, the difficulty of matching the realism of rich mobile contexts in laboratory settings makes field studies paramount because they can unearth unexpected behaviors and adaptations [180]. For example, Benford *et al.* coordinated one of the most extensive field deployments of a ubicomp technology, Can You See Me Now?, a mobile game in which participants raced through city streets to catch virtual avatars that they tracked on mobile devices [15]. The authors compiled data from a variety of sources, including videos, interaction logs, voice and text communications between players, and interviews, and found that connectivity and location tracking irregularities could be co-opted and integrated into the game as a *feature* rather than an error. Halloran *et al.* also showed that long term, situated design can promote a sense of ownership of the technology amongst stakeholders, encouraging adoption [63]. However, Zhang *et al.* and Davies *et al.* note that researchers find it difficult to make use of real devices and to observe and collect data in the field [180, 40].

	Replayer [129]	Context ESM [84]	Interview Viz [173]	Me [53]	iESP [34]	Mobile probes [78]	Rememberer [51]	d.Tools [65]	Active capture [30]	SUEDE [102]	DART [110]	Topiary [107]	Crossweaver [165]
Unified client system													
Client supports both needfinding and prototyping													
Leverage existing devices													
Mobile phone support				■		■							
Communication with external applications								■					
Multiple, lightweight communication options													
PDA application		■			■		■				■	■	■
Mobile application				■		■							
SMS/MMS support						■							
Qualitative, quantitative, and contextual data													
Client diary support						■	■						
Elicitation interview support			■										
ESM				■	■	■							
Event-contingent ESM		■		■									
Multimedia	■												
Location	■			■							■	■	
Proximity	■						■				■	■	
Logging	■										■	■	■
Not require fully implemented applications													
Mobile ESM and diary without installation						■							
Wizard of Oz support								■	■	■	■	■	■
Support the full experimental lifecycle													
Needfinding	■	■		■	■	■	■						
Lightweight prototyping								■	■	■	■	■	■
Integration with functional prototypes for summative studies								■					
Support monitoring and notification													
Real-time information visualization for experimenters	■					■		■	■	■	■	■	■
Real-time notifications to participants						■							
Support lengthy, remote studies													
Rule system allows complete automation (in some cases)								■					
Support for swapping remote wizards mid-study													
All experimenters can be remote from participants		■		■	■	■	■						■
Support coevolution													
Easily branch applications								■	■	■	■	■	■

Table 4.1. Tools compared by their support for different aspects of ubicomp experimentation supported by Momento.

From a tools perspective, many researchers have developed systems that support the rapid creation of throw-away functional prototypes to be used during iterative design (*e.g.*, Table 4.1). These tools often rely on Wizard of Oz so that fully-implemented applications are not required, support monitoring of events during evaluations, and allow developers to branch code. For example, Li *et al.* developed Topiary, which can be used to prototype location-enhanced applications in which a wizard typically shadows a participant and enters location information as the participant moves around [107]. Dow *et al.*'s DART extends Macromedia Director to allow designers to Wizard of Oz augmented reality applications. It provides multimedia support for video, tracking, and sensor data [46]. Klemmer *et al.*'s d.Tools explicitly integrates support for evaluation into their prototyping platform by showing the design of a task and the user data derived from completing that task together in one interface [102]. Chang and Davis developed Active capture to allow designers to prototype systems that direct human action [30]. Klemmer *et al.*'s SUEDE tool allows designers to prototype prompt/response speech interfaces [102]. Sinha and Landay developed Crossweaver to support rapid development of multimodal applications [165].

Several researchers have developed systems designed to gather or visualize qualitative, quantitative, and contextual data to support lengthy, remote studies via a lightweight communication infrastructure. Intel Research developed an early experience sampling tool, iESP, that ran on mobile platforms but was not networked [81]. Intille *et al.* developed applications to support context-based, or event-contingent, experience sampling [85]. This system runs on PDAs and can be used to support remote needfinding studies. Froelich *et al.* built the Me (My Experience) tool to gather a wide range of experience sampling data [53]. Their system runs on a Windows mobile platform. Morrison *et al.* implemented Replayer, a system that consolidates a variety of streams of quantitative and qualitative information into one interface to support mobile application evaluation [129]. Van House's InterviewViz is a desktop tool to support photo elicitation interviews [173]. Hulkko *et al.* ran a diary study using Mobile Probes, a personal publishing system that automatically converts mobile media messages to feeds that can be viewed by feed readers or browsers [78]. In the study they ran using the tool, the experimenters found that the ability to monitor participant captures as they occurred was critical. Also, they reported high rates of media capture, but the experimenters noted that the participants were experienced mobile phone users. Finally, Fleck *et al.*'s Rememberer tool allowed museum visitors to capture photos and other data about their visit at a particular exhibit by simply waving a card over a reader [51].

A handful of other projects have taken our approach of using mobile phones to enable realistic studies (*e.g.*, [71, 34, 8, 62, 80]). For example, Okabe *et al.* ran a diary study using mobile phones utilizing blogging software [135]. Specifically, the experimenters gave participants phones with a preinstalled application that submitted photos and GPS location data to a web blog. Experimenters were then able to use captured information during elicitation interviews. Also, Intille *et al.* designed a system for photo-based ESM, but it relies upon an infrastructure that would be difficult to implement in everyday field settings and involves context-aware capture rather than participant capture [83].

Recent work has suggested guidelines for tools that support experimentation. For example, in an analysis of several field evaluations of ubicomp applications, Crabtree *et al.* elaborated the need for tools that support side-by-side reviews of video and log data; mechanisms to support user captured video; synchronization between recordings; a timeline-based visualization interface for rapid review of different data sources at different times; and better support for annotation [37, 15].

4.5 Summary

While many of the systems we reviewed meet some of the requirements for ubicomp experimentation, none provide integrated, usable support for experimentation. In the next chapter, we introduce Momento, which integrates multiple experimental methods, and includes a collection of novel features ranging from privacy sensitive review to live monitoring and feedback that were motivated by our user-centered, iterative approach.

Chapter 5

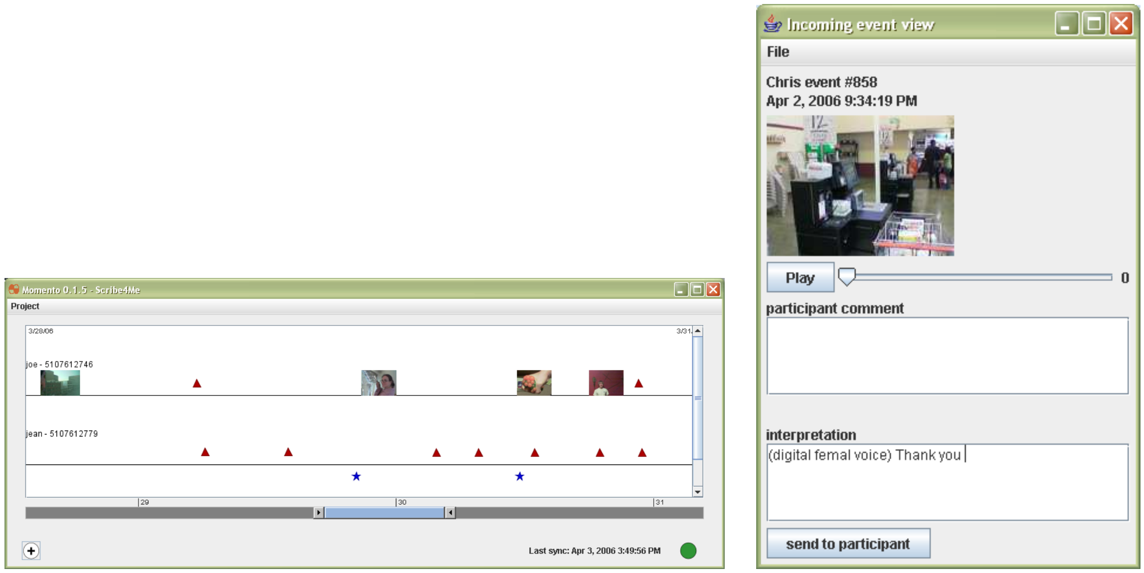
Momento architecture

Based on the implications we derived for supporting needfinding and prototyping tools, we developed Momento, a set of software tools that support early-stage evaluation and prototyping for ubiquitous computing applications.[1]

Momento consists of a set of configurable tools that can be used without writing source code. As shown in Fig. 5.2, the most important components of Momento are the clients (C) used primarily by participants to send messages and make requests (clients include fixed applications and mobile devices running either standard mobile multimedia applications or a specialized mobile application we built); the desktop platform used by experimenters to configure and monitor experiments (D); and the server (S), which supports multiple experiments, handles communication between clients and the desktop for each experiment, and provides data backup and remote access facilities.

All communication in Momento is sent as text or multimedia *messages*. Messages are sent to and from the server and mobile devices or fixed applications using HTTP or SMS/MMS (text or multimedia messaging). Momento communicates with networked applications using the Context Toolkit (CTK) [43]: messages are sent to applications using the CTK event system and are sent back to Momento using the CTK services system.

[1] This chapter is based on [28].



(a) (b)

Figure 5.1. **(a)** Part of an overview display for use by remote experimenters while a study is running. This display is especially valuable during lengthy studies with sparse data. **(b)** Part of a detailed view of a message.

5.1 Clients

Participants, not experimenters, are the primary users of clients. The most basic client facilitates *data gathering*, such as requesting a photo or implicitly logging sensed *context*. It is also possible to *simulate an interactive application* through the mobile client. *Mobile experimenters* can also use clients to report context information from the field or to interact with participants while mobile.

Data gathering The main purpose of Momento is to help support situated experimental data gathering. An important part of this is qualitative reports from users. In addition to raw SMS/MMS, Momento includes an easily configurable mobile client that can be installed on most mobile devices. This client can display an information request to a user, who can then respond by taking a photo (if supported by the device), recording audio, entering text, or sketching using a stylus.

Context Both the mobile client and fixed applications can report on context data. The mobile client supports logging of location, nearby people and audio.

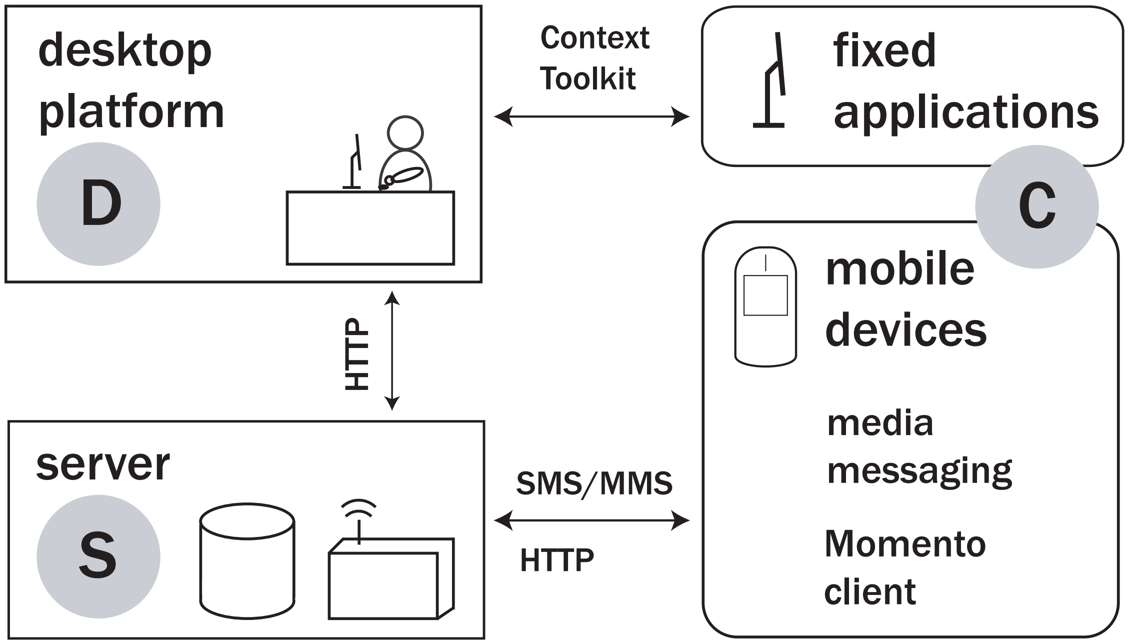


Figure 5.2. System architecture. The desktop platform (D) and server (S) communicate with clients (C) via SMS/MMS, HTTP, or the Context Toolkit. Study data is stored for later analysis or retrieval via the web.

Simulating applications In addition to information requests, arbitrary messages may be sent to clients independently or in response to user input. This can be used to simulate an interactive application. By default, the Momento mobile client displays incoming messages to the participant (more sophisticated behavior must be hand-coded, as must any behavior by fixed applications). Also, the client may be configured to display one or more buttons that a user can select. Depending on its configuration, a button may send locally cached data to the desktop platform or may request media from the user and send it (*e.g.*, ask the user to enter text or take a photo).

Mobile experimenters Although we support experimental protocols that do not require an experimenter to be mobile, there are still occasions when mobility is desirable. For example, a mobile experimenter might wish to observe participants or to record data that is hard to sense. Mobile experimenters in Momento can use the mobile client to report data or to view live data recorded by participants. In particular, messages can be forwarded to a mobile experimenter who can optionally respond to participants via Momento. Finally, Momento can notify an experimenter when a participant is in a certain context (*e.g.*, in a certain location with a group of people). The experimenter may wish to act on this information, for example by making additional observations in person.

The supported version of the Momento Mobile client is implemented in C#. The mobile client can receive messages via SMS/MMS from the server and can send messages back using either SMS/MMS or HTTP over a network connection, if one is available. The client expects to receive messages of the following format: `header|participantId|study|eventNumber|text[,buttonName,buttonName...]`. The header string is a flag for messages coming from Momento and is discarded. If no button names are sent, the client replaces the description field text with the 'text' string. However, if any number of button names follow the text, then the client assumes that this message is a question description. In this case, the client creates a new form using the 'text' string as the question text and displaying the appropriate buttons ('buttonName' should correspond to a name field in a button description in the mobile configuration file). In this case, when a question is answered the client reverts back to the main screen and the description text is unchanged.

In this way, clients on mobile devices can leverage the fastest network connection available to them (*e.g.*, 802.11, GPRS, TCP/IP over Bluetooth, *etc.*) to send messages to the server, which will pass them on to the experimenter's desktop platform.

The mobile client is configured using a simple text file. Listing 5.1 shows the basic structure of the configuration file. The experimenter specifies: 1) A description of the study that can be any string and appears once in the file; 2) An identification of the participant that should match the participant identification in the desktop platform and appears once in the file; 3) An IP address for the server that appears once in the file; 4) Up to four different data types (audio, Bluetooth, photo, or gps) to be continuously buffered as well as a one of four buffer lengths for each data type; 5) Any number of button definitions, where a button definition includes the name of the button, an optional color value for the button, the text label for the button, a switch indicating that captured photos can be annotated (default is no annotation), switches indicating that manual photo, audio, and text capture should occur when the button is pressed (default is no manual capture), a switch indicating that a buffer

Listing 5.1. The structure of the client configuration file

```
# 1 Description of study (Any string)
description, Scribe4Me study
# 2 Participant ID
id, 123456789
# 3 Server location (IP)
serverLocation, 236.345.2.1
# 4 Continuously buffered data types
# (name, buffer length in secs, and automatic uploading)
# bufferAudio|bufferBt|bufferGps [30|60|...] [true|false]
bufferAudio, 30, false
# 5 Button definitions:
# button name=name, label=labelText,
# [color=color], [defaultScreen],
# [manualPhoto|manualVideo=secs|manualAudio=secs|manualComment|
#  sendAudioBuffer|sendBtBuffer|sendGpsBuffer|annotatePhoto]
button, name=what, label=what happened?, defaultScreen,
sendAudioBuffer, manualPhoto
```

with the appropriate data type should be sent when the button is clicked (default is not to send any buffered data), and a switch turning off photo repositioning when annotating.

Listing 5.1 corresponds exactly to the Scribe4Me application (Fig. 2.1) [120]. Audio is stored in a 30-second buffer, and the interface contains a single button (labeled "what happened?"). When the participant presses the button, the application asks that a photo be taken and then transmits the photo and current audio buffer to the experimenter's desktop platform.

5.2 Desktop Platform

The desktop platform is the main interface used by experimenters. It supports many of the activities of the experimental lifecycle. Experimenters can use it to manage the *specification* of the devices, participants, groups, locations, and rules associated with the experiment. While the experiment is running, experimenters can see an *overview of communications* with the participants that supports peripheral monitoring, *manage communications* by sending messages and viewing incoming messages, and *receive notifications* about actions that are needed. The desktop platform sends text and media messages to mobile devices with the server's help, and communicates with other client applications via the Context Toolkit. All incoming and outgoing messages are stored locally by the desktop platform. Experimenters may upload this data to the server at will. Here we discuss each of the activities that is supported.

Specification of the experiment Specification, which typically takes place during or just after the planning phases of an experiment, includes the devices, participants, groups (of participants), locations, and rules.

Participants and groups Participants are uniquely defined by their client ID (typically a mobile phone number). Other information about participants can be included, such as: their email address, a designated place, capabilities of the mobile device (whether or not it has MMS and whether or not it has the mobile client installed), notes, and a calendar schedule (see Appendix A for details). Experimenters can create groups of participants for use when rules or messages should apply to more than one person. Experimenters can create any number of groups and can add a single participant to any number of groups.

Locations Momento supports both discrete semantic locations (*e.g.*, within discovery range of a Bluetooth beacon) as well as geographically defined locations (within a bounding box specified by two GPS coordinates). GPS coordinates are sensed via a Bluetooth connection to an external device.

Rules Experimenters can pre-configure outgoing messages using Momento's rule system. The rules system can be used to automate simple, frequent actions and to configure ESM and event-contingent ESM studies. Momento evaluates rules in a continuous loop on a separate thread. When configuring a rule, experimenters define rule conditions (what fires this rule?) as well as rule actions (what to send and to whom). Rule conditions include participant location, proximity, incoming message pattern matching, as well as time-of-day. Rule actions include sending files or text. Note that when a folder is specified, Momento randomly selects a file from that folder. Also, Momento supports some special macros in the message text when responding to an incoming message: %%PARTICIPANT will be automatically replaced with the participant's name, %%LOCATION the participant's location, and %%PROXIMATE the other participants near this participant. These conditions and actions can be combined using simple booleans. Rules generally take the form: `if` [conditional and/or conditional] `then send` [content] `to` [recipient(s)] (see Table 5.1). So, for example an experimenter could configure rules that 1) check to see if participant Alice is in her office at noon and every half-hour thereafter, and if so, sends her an image file (see Figure A.3 in Appendix A); 2) sends a message to group "wizards" whenever a message from participant Bob arrives; 3) sends Alice a list of the people near her whenever she sends Momento a message that matches the regular expression `(W||w)here am (I|i)?`; *etc.*

Overview and management of communications The desktop platform's overview mode, shown in Fig. 5.1 (left), displays recent messages sent to and from all participants. The timeline, built using Prefuse [66], associates one horizontal line with each participant. Red triangles, or photos (if part of the message), represent incoming messages, while blue stars represent outgoing messages. A vertical red line indicates the current time. Blue stars shown past this line indicate future messages scheduled using the rule system. Clicking on any of the message icons opens a message information panel showing message details and any associated media (see Fig. 5.1, right).

Notifications Notifications are provided automatically when messages arrive to help experimenters monitor incoming messages. Notifications can allow an experimenter to multitask and can reduce the impact of experimenter fatigue. They

if [cond and/or cond ...]	*then send* [content]	*to* [recipient(s)]
o/g near/!near location	file (random | specific)	any person
o/g arrives/leaves location	[o's name]	any group
o/g stays in location (min time)	[o's location]	
msg arrives	[o's proximate people]	
msg has attachment	[incoming message]	
msg matches regular expression	arbitrary text	
time of day (repeat time)	web page	

Table 5.1. Rules. o = originator of the message; g = group. In the conditional, g is matched if anyone in the group is the originator. Momento evaluates rules continuously (on a separate thread).

are especially helpful when data are sparse. Notifications are visual and auditory.

5.3 Server

The Momento server has three functions: it acts as the *gateway* between the desktop platform and the cellular network, (allowing the desktop platform to run on any networked machine); it *manages study data*; and it provides password protected *web access* to captured media. Experimenters can configure a server using an Administration program 1) to create a Momento archive on a machine; 2) to create users and studies, and 3) to associate users and studies.

Gateway The server interfaces can communicate video, images, audio, or text-based data to and from email addresses, default messaging applications on mobile phones, or the Momento mobile client. The server automatically compresses media, and sends messages to phones via a GPRS-enabled modem. When sending messages to mobile clients (rather than email addresses), the server relies on an SMS/MMS gateway to communicate with a cellular modem. Currently, the server is configured to support standard installations of the NowSMS Gateway (http://www.nowsms.com/). NowSMS supports a variety of cellular modems.

Study management The server stores all data associated with each study. All data are stored in a folder with the same name as the study. Experimenters can use the desktop platform to upload data to and download data from the server by specifying server information, a study name, and a user name and password associated with the study (the association would be specified using the Administration program specified above). Note that the server is capable of storing any file type, including images, audio, and video files sent to and from the desktop platform.

Web host Web access is hosted by the server but can be controlled with the desktop platform (see Appendix A for web host configuration information). Data for each participant is provided separately, and a per-participant password (provided by the experimenter) is required for access. The web interface is implemented using PHP (see Figure 5.3).

5.4 Example

As an example, consider a study we recently ran of the Scribe4Me system (shown in Fig. 2.1), which allows people who are deaf to request a transcription of recent audio. The goals of our experiment were to learn when and whether such functionality would be valuable to people with a range of hearing impairments. For this reason, our experimenters wanted a photo showing additional context relevant to each audio request. Also, they used a diary protocol to ask participants to answer a fairly lengthy series of questions about each transcription request. This was done at the end of each day. Although our study ran with an earlier version of Momento, here we describe how it would be done with the current version.

Because of Momento's ability to capture live context, including audio, and transmit it to experimenters, Scribe4Me requires no implementation, only configuration. In general, we have found this to be true of many studies that involve an application that can be defined simply by transferring information and media between participants and a human experimenter. The participant device is configured to buffer 30 seconds of live audio at all times. Additionally, a button, labeled "what happened" is added that transmits that buffer to experimenters when pressed. This button also

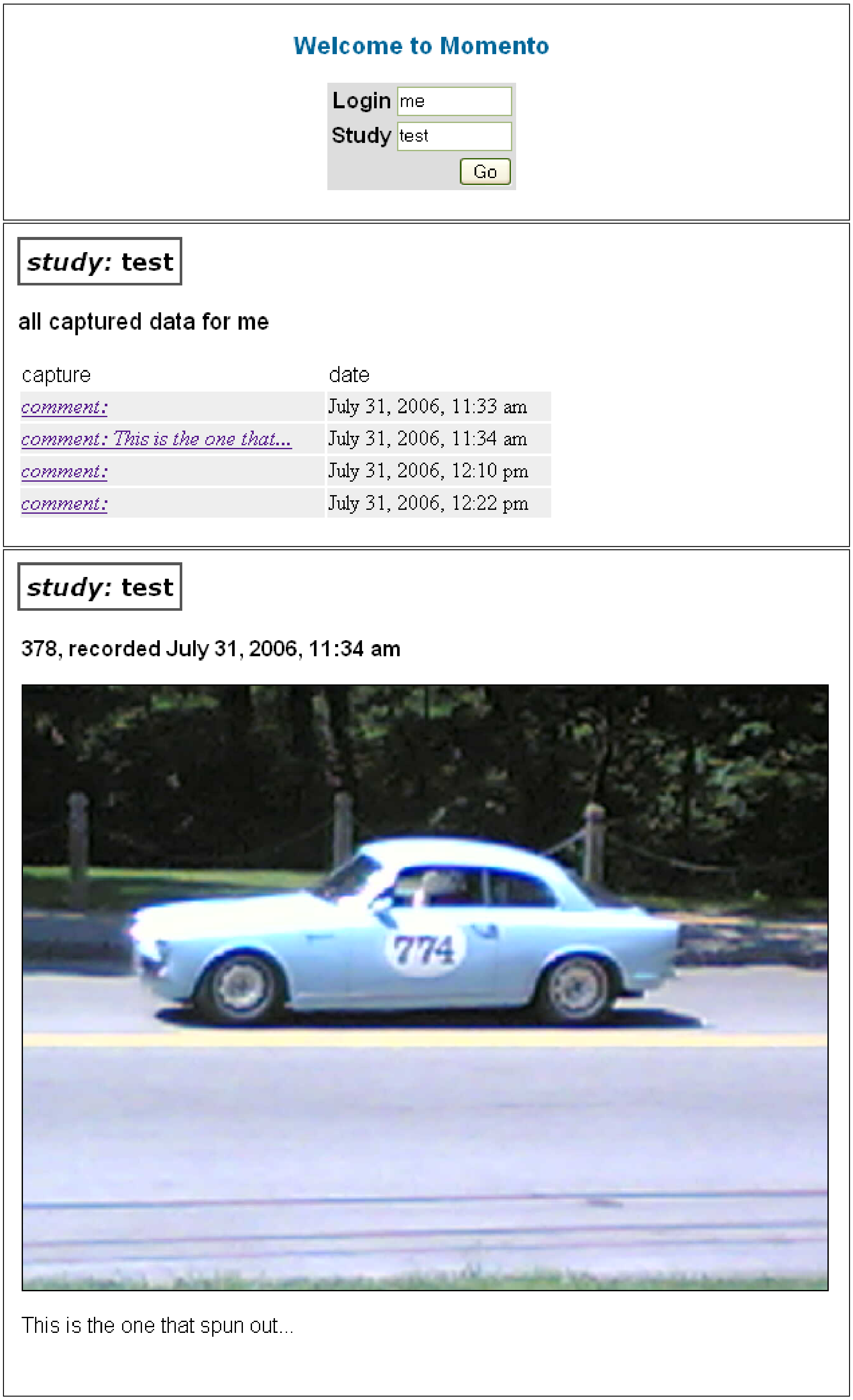


Figure 5.3. The web host login, capture list, and individual capture access screens.

causes a pop-up window to appear, asking participants to take a photo illustrating the reason for the request.

To run this experiment with Momento, only 4 actions are needed. First, the experimenter downloads and installs the mobile client on each participant's mobile device. This is a standard mobile device application installation process and involves moving the Momento client archive file to the device (via Bluetooth, serial port connection, docking station, *etc.*). Once the file is on the device, the experimenter only needs to double-click on the archive file to install the application.

Second, the experimenter modifies a total of five lines of the configuration file (as shown in Listing 5.1) to indicate the phone number of the Momento server and the participant's unique ID, to display information about the experiment to the user, to buffer audio, and to add an action button to the mobile client.

Third, the experimenter configures the desktop platform, specifying the IP of the server, the study name and associated user name and password (see Figure A.1 in settingsConfig) and the phone number and ID of each participant (see Figure A.2 in Appendix A).

Fourth, the experimenter can optionally specify details about what context should be gathered (by editing the mobile client's configuration file) or add rules to the desktop platform describing when communications should take place automatically (the rules panel is shown in Figure A.3 in Appendix A). For example, the experimenter could change the configuration file to indicate that Bluetooth data be gathered automatically, or that the desktop platform should send a message to participants every hour. The experimenter may also add information about a mobile experimenter to the desktop platform's configuration. The experimenter must also add a rule for each type of data that mobile experimenters should see and respond to, and a rule that transmits any incoming information from mobile experimenters to the participant as needed.

Once configuration is complete and the desktop platform is running, event flow would look something like the following. Note that all communications from the participant to the experimenter go through the mobile client, to the server, to the desktop platform and all communications from the experimenter to the participant do the reverse, unless otherwise specified.

1. Scribe4Me participant presses "what happened"; sends audio and photo to experimenter.
2. Experimenter sees data and enters transcription, which is sent to the participant.
3. At the end of the day, the participant reviews all of her requests using a web interface provided by the server, and fills out a form answering a series of questions that are then sent to the experimenter.
4. Optionally, an experimenter could also send a short-answer ESM request to the participant immediately after completing a transcription or at random times during the day. The participant's response would be shown immediately to the experimenter.

5.5 Summary

We developed Momento to meet the requirements we derived for supporting needfinding and prototyping tools. In the next chapter, we explore how Momento can be used.

Chapter 6

Momento validation

In this chapter we provide more detailed examples of how Momento can be used: Experience sampling, diary studies, and Wizard of Oz prototyping. We also describe experiments run using Momento.[1]

6.1 Uses

6.1.1 Experience sampling

ESM studies intended to reach large numbers of SMS-savvy users can be run with no installation of the mobile platform. In this case, periodic queries can be configured using the rule system of the desktop client. Queries are delivered and responses (including text and/or photos) are gathered using text or media messaging. When the mobile platform is installed on participant phones, queries can be context sensitive (for example, triggered by a change in location).

[1] This chapter is based on [28].

6.1.2 Diary studies

Researchers conducting diary studies usually give participants a high-level query, such as "chronicle important events in your life that have to do with your family," Participants are then responsible for recognizing and chronicling events relevant to the query. Diary studies are a good way to gain insight into the attitudes and beliefs of participants, but, because of their open-ended nature, require a degree of monitoring by researchers. Again, they can be used either to gather requirements or to understand the use of a deployed application.

As with experience sampling, researchers can run diary studies without requiring any installations on participant phones. Since in diary studies participants are responsible for detecting important events, researchers do not schedule queries, but participants can send captured data *via* media messaging.

When the mobile environment is installed, Momento can supply some reminder and feedback information to participants. Earlier investigations of diary studies revealed that it is important for participants to be reminded of the types of events they are to capture [156]. To this end, Momento displays a persistent description of the types of events that participants are to capture.

Momento is ideally suited to support the diary study pipeline described in Section 3.3. Experimenters can monitor participant captures and annotations and provide feedback if necessary. All incoming events are logged, allowing researchers to review them at any time. Also, when participant responses lack detail or become less frequent, researchers can send feedback or requests for additional information directly to participants' mobile interface using the desktop platform. Finally, the web history supported by the server, as well as the navigation options in the desktop platform, can be used to facilitate elicitation interviews after a few days of diary data have been collected.

6.1.3 Rapid Prototyping of applications

Our Scribe4Me scenario described a simple Wizard of Oz application implemented using Momento. Momento has integrated support for Wizard of Oz prototyping, including tools to visualize and notify the wizard about incoming events. The wizard can pre-configure responses to participants or enter them interactively. Rules can also be configured to support simple data retrieval applications.

Alternatively, more complex prototypes may require coding. Momento is designed to support this in two ways. First, it is integrated into the Context Toolkit. Anything that Momento can sense or that users report to Momento can be sent to applications using the Context Toolkit. Second, mobile developers can use mobile prototyping environments (such as Mobile Processing ™(`http://mobile.processing.org`)) to create more complex mobile clients that communicate with the desktop client *via* SMS/MMS or an HTTP connection. The advantage of integrating these clients with Momento is the ability to depend in part on a wizard, and all of the features of Momento which support logging and evaluation.

In general, Momento best supports prototypes that have some mobile component, either in the prototype itself or used by the wizards to support the prototype. Alone, Momento is capable of only supporting the most rudimentary capture and access, mobile-based applications. Coupled with context-aware applications or data visualization interfaces, though, Momento can play a role in wide variety of prototypes by providing a clearinghouse for automatically- and manually-sensed data.

6.2 Other uses for Momento

Momento could also support probe based studies [146, 79, 67]. Text and or images inspiring thoughtful, situated responses could be sent to participants at opportune times. The Momento client could also be configured to support capturing information for such studies.

Momento can be used to "fake" sensing capabilities by asking human participants to act as sensors when information is difficult to sense or sensors are hard to deploy. This has been done successfully in the past for simulating location information [16], and we used this technique to scaffold the AwarenessBoard application (our study of that application is described later in the document). This capability might also be used to reduce ambiguity in sensed information. For example, a Context Toolkit sensor could send an event to Momento to have a human resolve ambiguous data.

Finally, Momento can be used to notify experimenters about critical events. For example, a rule could specify that a message be sent to an experimenter's mobile device each time someone interacts with a public display. Rather than having to spend all day watching the public display, the experimenter could simply walk over to it when receiving that notification, and take notes or ask the user to answer some questions.

6.2.1 Summary

Momento supports situated ubicomp experimentation by managing, recording, and automating different types of data flow between participants, experimenters, and applications. Momento was developed in an iterative, user centered fashion. Our validation was conducted as part of that iterative design process. We deployed our system internally and with external experimenters as we developed it. Each time, we interviewed experimenters and end users, and analyzed the data that experimenters gathered. Our focus was on understanding and improving both the end user experience and the experimenter experience.

6.3 Experiments

Table 6.1 shows an overview of the five experiments run using Momento. The experiments included: a pilot; a public awareness display (AwarenessBoard); the Scribe4Me system; a diary study of young adults' approaches to informal learning (InformalLearning); and a mobile sketch-based learning application for children (PhotoSketch). These experiments demonstrate Momento's ability to support a variety of methods, data collection techniques, and a range of study lengths. The pilot and AwarenessBoard studies were run with an early version of Momento, while Scribe4Me was run about five months later with an intermediate version, and the other two were run when Momento was nearly in its current state. We iterated after each one.

Study name	Study type	Number of participants	Days per participant	Number of experimenters	Wizard of Oz	Feedback	Questions	Desktop review	Other applications	Mobile experimenters	Distributed experimenters	Desktop real-time	Mobile client	Rules
Pilot	field study	4	1	4	■	■	■					■		■
AwarenessBoard	field study	14	51	3	■		■		■	■	■	■	■	■
Scribe4Me	field study	6	7-15	2	■					■	■	■	■	■
Informal learning	diary	12	7	4		■		■			■	■		■
PhotoSketch	workshop	24	1	7				■					■	

Table 6.1. How experimenters used Momento in five different experiments, including whether or not the experiments involved Wizard of Oz, feedback or questions sent by experimenters to participants, use of the desktop to review captures with participants after the experiment, external applications, distributed or mobile experimenters, monitoring of the desktop during the experiment, the mobile client, or rules.

The experiments spanned a range of methods and experimenter conditions. The pilot, AwarenessBoard, and Scribe4Me studies were deployed as field studies lasting days to months, while PhotoSketch was deployed during 1-day workshops and the InformalLearning study was a week-long diary study. Scribe4Me depended heavily on synchronous exchanges between experimenters and participants, while AwarenessBoard and the InformalLearning study required only minimal exchange and PhotoSketch required none. Furthermore, experimenters in all experiments monitored captured events in real-time, but those who ran the PhotoSketch and InformalLearning experiments also used Momento for *post hoc* review with participants. Finally, in the AwarenessBoard experiment, Momento communicated directly with not only participants but also other applications.

In this section, we describe our experiments in detail. Our goal is to illustrate that we were successful in providing a usable experience to both end users and experimenters, and to illustrate the range of uses for Momento. For each experiment, we first provide a brief description of the experimenters' goals and describe the set up for the experiment. We then describe what we learned about both the participant and experimenter experience and how this led us to iterate on the design of Momento.

Figure 6.1. A wizard monitoring Momento while focusing on another task.

6.3.1 Pilot study

We ran a pilot study to evaluate the user interface and give us insights into how Momento affects both the experimenter and participant experience. We asked experimenters to run a Wizard of Oz study focusing on two types of information: location and traffic conditions. We provided the wizards with participants and a predefined protocol for the study. Wizards were instructed to use Momento to query participants about their location via ESM. Each participant was told to query the wizard for location data about other participants (*e.g.*, "Where is Steve?" or "Who is in the lab?", *etc.*) or traffic information (*e.g.*, "How is the traffic on the bridge?").

Because this was the first time we observed this technique in use we also wanted to give participants and wizards a chance to brainstorm actively about other possible uses for the system. To encourage this open-ended use, we instructed wizards that they could come up with other experience sampling questions and participants that they could query the system for any information they need throughout the day.

Setup

To run the study, we recruited total of four experimenters with at least an upper-level undergraduate understanding of user studies. For one working day, two of the subjects played the role of participants, while two switched off between being participants and wizards. Momento is designed to be used as much as possible in the *context* of other work. Thus, wizards were asked to use Momento from their personal desktop machines (Figure 6.1). The job of wizard was traded off during the study, with each wizard's turn lasting at most 2 hours, followed by a break of 1 to 2 hours. We followed up with open-ended interviews with each experimenter.

Participant/experimenter experience

We found that the system was intuitive to use and were inspired by the range of uses for the system in the open-ended section of the study. Overall, during the one day study, wizards generated 55 messages and participants sent 58 messages. However, only a small number of those were related to the main questions of the study. There were four traffic requests from participants and four responses to those requests. Also, there were five location requests and five responses to those requests and total of seven questions generated querying participants' location (five scheduled events and two automated events).

The rest of the messages were related to uses of the system that wizards innovated. For example, one wizard setup questions that participants could respond to with scalar results. One of these questions was related to an event that occurred during the course of the study:

> Is the current [lab] tour distracting your work? Please reply on a scale of 1 for not distracting, 5 for impossible to work

Other questions were more general, such as:

> How busy are you right now? 1 not busy at all, 5 is really busy.

Also, wizards used the system to compensate for their sometimes not being able to gather information as rapidly as an interactive system might. For example, the wizard used the automatic reply feature in Momento to respond with the following text whenever a participant sent a request with text that matched the regular expression ".*traffic.*":

> I'll check the traffic for you as soon as I can and get back to you

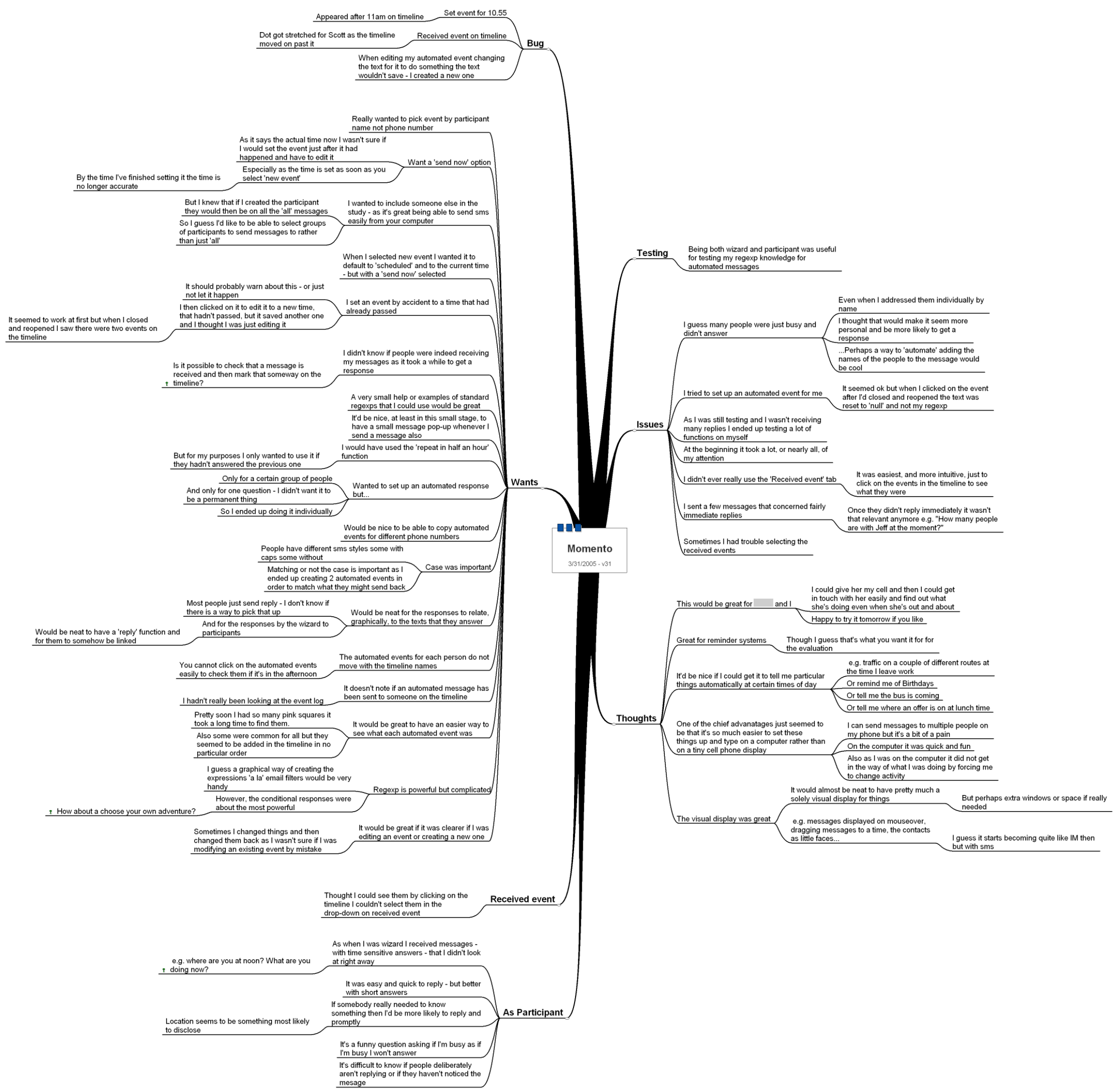


Figure 6.2. A brainstorm map created by an experimenter in the pilot study using MindMapper®software.

There were also more playful uses of the system. For example, one wizard configured the following two messages: one a question and one a response if a participant sent the correct answer.

> What is the airspeed of swallow?
>
> African, I mean American, aaahhhhhhh.

One wizard extended the idea of prototyping games using Momento and created a short text adventure game during the study. The first message the wizard sent was:

> You're in a forest. There is a trail to the N and a cabin to the W. Which way would you like to go? N/W

The wizard then set up two automated messages that would fire based on participant responses. If a participant sent back a string that matched the regular expression "W.*", Momento responded with:

> You open the cabin but there was a troll and he squashed you with his fist. Game over. Bad luck

If a participant sent back a string that matched the regular expression "N.*", Momento responded with:

> You followed the trail and there was a big pot of gold and a beautiful maiden. Congratulations, you win!

Interviews with experimenters revealed a high-level of interest in the system. During one interview, an experimenter brainstormed potential uses for the system using brainstorming-support software he uses on a daily basis (Figure 6.2). Experimenters reported that using Momento was not too distracting and "felt like e-mail or instant messaging." One experimenter suggested modeling the interface after instant messaging, adding clickable "faces" for each participant. Also, there were a set of features that experimenters thought should be added to the interface. The most pressing need reported was a means of visually linking questions and answers. At the time, the timeline visualization showed generated and received messages but did not indicate the relationship between messages (*e.g.*, whether a message was a response to a previous message, or which message it was a response to). A similar issue was that experimenters wanted to schedule a message to repeat only if a question had not been answered. Also, experimenters wanted the ability to create groups of participants. While Momento supports sending a message to all participants, experimenters wanted to send messages only to subsets of participants. Experimenters also thought there should be a "Send now" button on generated messages. Finally, experimenters sometimes had difficulty selecting overlapping message icons in the timeline visualization.

Implications for design

Based on feedback from this study, we made several changes to Momento, including:

- Added a notion of places and groups to the desktop platform to support social-mobile applications.
- Expanded the rule system from AND-only to AND/OR. This allows more sophisticated sampling and prototyping without sacrificing much simplicity.
- Began development of a J2ME mobile client to expand range of supported applications that can make use of data streams not available via standard media messaging, such as Bluetooth.
- Added table view of rule and action events. Before, these items were difficult to locate in the interface – small items on the timeline display.
- Augmented the desktop platform logging system to include all experimenter and participant events. The pilot made clear that a complete log of events would be critical for any field study.
- Rewrote the timeline display using Prefuse [66]. The timeline display visualizes all messages sent and received by the desktop platform. Prefuse made it easier to show relationships between events (*this* message was a response to *that* question) and swap representations of messages (*e.g.*, a message could be represented as a photo or a shape).

Overall, while this study was small and made use of only a subset of Momento's capabilities (*e.g.*, we did not use MMS), the interface seems to fill a need for mobile application developers. Our experiment showed that the system works well for standard applications, such as traffic and location information, but also that there is a potentially large set of additional applications that could lend themselves to prototyping using Momento, especially with the above additions. The success of the open-ended section of our study also suggests that it may be valuable for developers to take advantage of "active brainstorming" using Momento to generate ideas for applications.

6.3.2 Early study of AwarenessBoard

We used Momento to help implement an early prototype of a public display, the AwarenessBoard (shown in Fig. 2.2), intended to convey a history of the availability and location of participating faculty members in our department. The AwarenessBoard study was one of the earliest and most extensive studies run with Momento, and it contributed greatly to our understanding of the usability issues facing both participants and experimenters. It depended on almost all of Momento's features. The study lasted for two months and involved fourteen participants (including 12 faculty and two students). We were the primary experimenters, although we were collaborating with two social scientists who had designed an application to help test sociability and awareness in a distributed academic department.

Setup

The public display was designed to show faculty members' location availability, which was provided by Momento using the Context Toolkit. The mobile client was customized to show the same information. Sensing of location and availability was done using Bluetooth beacons in Momento. Availability was "sensed" using a simple heuristic that leveraged location information: If in a public place or alone in his or her office, a faculty member was available. If others were present, a faculty member was assumed to be in a meeting. Changes in sensed availability triggered SMS messages to faculty checking whether the estimation was correct. Faculty could optionally provide a calendar indicating times they did not wish to receive such interruptions. Experimenters monitored the desktop platform to ensure faculty were answering questions in a reasonable amount of time. The availability heuristic and location sensing were implemented using Momento's rules system.

In total, we recruited twelve faculty participants to use the public display and answer questions using the mobile client. Faculty were each given mobile phones so they could respond to SMS messages confirming estimations of availability. There was too much variability in the technology they already carried to piggyback on existing devices. We also recruited two students who tested an SMS interface to the public display. These participants could send a message to Momento requesting the current availability of a faculty member. Responses to these requests were handled by a wizard monitoring the Momento desktop platform.

We ran pilots of the system for three weeks and ran the study over the course of two months. We iterated and improved both the public display and the client based on participant feedback over the course of the study, and made significant adjustments to the public display after the pilots and again during a holiday break six weeks into the main study.

Participant experience

Our participants were biased towards a population that rarely if ever use phone-based applications or SMS/MMS, and this led to some difficulties with the mobile client. Additionally, the unfamiliarity of the particular phone we gave to participants led to complaints that the phone was "big and bulky" or "stopped working" (the operating system on the phone we used notified users of irrelevant information and used modal dialogs to demand confirmations). As a result, users did not always keep the phone nearby.

For these reasons, it was difficult to derive useful feedback for iterating the mobile client's interface. Primarily, this experience encouraged us to focus on improving our mobile client to better piggyback on mobile devices that individual users already carry.

Experimenter experience

Regarding setup, implementation of the application being tested (shown in Fig. 2.2) was time consuming. However, linking it to the Momento infrastructure was straightforward. A bigger difficulty arose when setting up security for the Java-based mobile client during mobile device installation (please see Appendix C for more information). To address this, we began working on a C# mobile client.

The day-to-day effort of running this field study varied. We found that experimenters controlling the mobile interface to the public display were able to monitor and respond to incoming questions with only minimal distractions from their other work.

Also, experimenters wanted to observe participants using the public display without necessarily always having to be present. To support this, they suggested having Momento automatically generate messages that would be sent to experimenters' mobile devices whenever it was detected that a participant was using the display. Experimenters also wanted to be able to send messages to participants once they had been in a particular place for a certain amount of time. We added these features partway through the experiment.

Implications for design

This study contributed significant information that fed into Momento's iterative design. Because this study involved many different experimenters working from different locations, we had to increase the sophistication of the server-desktop platform networking. Initially, Momento handed-off messages from the server to the desktop platform through a networked folder, which required the desktop to be on the same network as the server.

Part way through the study, we added support for experimenters to walk through the hallways checking for participant availability and sending updates to Momento via mobile phones. This and other unanticipated models for participation by remote and mobile experimenters led us to expand Momento's sender-recipient model (who could send messages to whom) and to enhance the rule system to support time-based triggers and triggers from external applications connected via the Context Toolkit. With this change, experimenters were able to tie events triggered by external applications to messages sent to mobile experimenters. Finally, we modified networking between the server and mobile client to leverage HTTP. By supporting HTTP uploading, we could avoid slow media messaging communication when other data carriers (*e.g.*, 802.11b) were available.

6.3.3 Field-based study of Scribe4Me

Scribe4Me, the example application described previously, was tested in a field setting using an early version of Momento. This deployment depended heavily on one experimenter monitoring the desktop interface, and included mobile devices running the Momento client. This two-week study involved six participants and was primarily conducted by a separate experimenter. However, that person was local to our institution, and we worked closely with her to provide ongoing support (to the point that we co-authored a paper on the results of the study [120]). Our goals in this study were to demonstrate how Momento could support a lengthy field study of a mobile prototype that depended heavily on experimenter support. The results of this study were used to help support the iterative design of Momento.

Setup

Experimenters used Momento to evaluate the Scribe4Me prototype described earlier and shown in Figure 2.1. The transcription service (provided by an experimenter) was available between 9am and 6pm. Participants were prompted to manually capture a photo of their surroundings each time they pressed the button (the photo provided helpful information for participant diaries written at the end of each day).

Participants were given new devices running the Momento mobile client. They were also asked to answer daily diaries. Because the web interface was not yet functioning when this study was run, an email was constructed by the experimenter each night showing a summary of participant requests, and sent to each participant along with a series of questions to be answered.

Participant experience

Participants were familiar with mobile devices – they used them on a regular basis and sent multiple messages a week – and they were able to use the Momento mobile client as it was configured for this study. The interface provided the core features users needed for the prototype: a simple method to submit requests (a single button) and a method of receiving transcriptions that was easy to access (text messages). The system worked well, and as a result users were enthusiastic about the prototype and used it in many experimental contexts, giving researchers valuable feedback about Scribe4Me.

There were two technical issues with the Momento interface raised by all participants: the delay (3-5 minutes) between requesting and receiving a transcription resulting from sending media messages across different operator networks, and the less-than-perfect quality of audio recorded (resulting in less complete and accurate transcriptions than a present observer could make). For three participants, the delay limited the scenarios in which they found the prototype useful, since they wanted a real-time aid for communicating with others. However, three users found the prototype to be valuable in a number of situations even with the delay, "It is a great idea. There are things that I'm curious about on a day to day basis. I don't mind waiting for the transcription." Momento now has support for leveraging WiFi networks when available, which reduces the delay to about 1 minute (listening and transcription time).

Also, because transcriptions were sometimes split into multiple text message due to the GSM network limit, users found longer transcriptions difficult to parse. The most recent versions of Momento address this issue by sending text directly to the mobile client using other protocols (*e.g.*, HTTP).

User requests were variable (see Figure 6.3). However, five participants used the application more-or-less continuously throughout the first week of the study, and two participants used the application for two weeks.

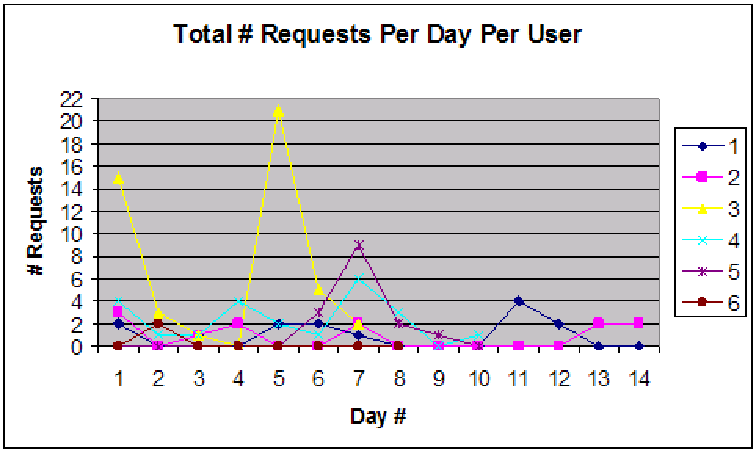


Figure 6.3. Total requests per day per user in the Scribe4Me study.

Experimenter experience

To reiterate, this study involved only one primary experimenter. The experimenter generally found it straightforward to configure Momento for this evaluation. The experimenter using the desktop platform had few problems reviewing and responding to participant requests. However, the experimenters did find it difficult to send responses to participant messages on the desktop platform. This process required multiple clicks and reselection of the participant to whom the message would be sent. Furthermore, experimenters complained that a modal dialog indicating that a message had arrived interfered with other work.

Also, delays in the cellular network made it untenable to rely on mobile experimenters for transcriptions. As a result, a single experimenter monitored the desktop platform 6am-9pm for the duration of the study. When analyzing the results of the study, experimenters were able to import logged data into spreadsheets but found that they would rather have representations of the data that could seamlessly associate participants with media that they captured.

Implications for design

Based on feedback from this study, we added a response field directly to the incoming message detail view so that experimenters could rapidly provide feedback to participants. We also removed all modal notifications so that Momento would better integrate with experimenters' work practice, and compensated by increasing the size of representations of new incoming messages in the timeline view.

Overall, Scribe4Me's evaluation was a success [120]. Scribe4Me also showcased the core features of Momento: media capture, experimenter support, and mobility. The main difficulties experienced by both experimenters and participants involved network delay, an issue that can be addressed by leveraging a wider variety of networking options, and that should be increasingly minor as networks continue to improve.

6.3.4 Diary study of informal learning

Four external experimenters used a recent version of Momento to conduct week-long diary studies of young adult's approaches to learning new technologies. The study was conducted at two sites: three experimenters ran six participants at one site and two experimenters ran another six at the other site (one experimenter worked at both sites). All participants were 14-19 years old. Also, across both sites participants started and completed the study on different days. We interviewed experimenters about the data resulting from their study. Our goals in observing this study were to get feedback on the ability of Momento to support studies distributed across different sites and multiple experimenters as well as back-and-forth interaction between experimenter and participants during situated needs finding.

Setup

Participants used standard media messaging applications on their mobile phones to capture images. Experimenters then conducted in-person interviews with participants, using the Momento desktop platform to review captured data. Later in the study, experimenters also occasionally sent feedback to participants to indicate that they needed to focus on capturing different events or simply to offer encouragement.

Participants unfamiliar with MMS were instructed to send text descriptions of events via SMS and, in some cases, given a digital camera. Experimenters shared data across different sites using the Momento server as an intermediary. Typically, one researcher would run the desktop platform for a couple of days before handing off the study to another experimenter. To do this, an experimenter would use the Momento desktop platform to upload the study to from her personal computer to the server, and a second experimenter would then use Momento to download the study from the server to his personal computer.

Participant experience

The study was in line with most participants' habits – they were comfortable taking photos with their phones. These participants were comfortable sending photos via media messaging to friends, and were able to use the same feature to send photos to Momento. This study helped to validate the advantages of piggybacking on familiar devices and applications.

Initially, participants were somewhat frustrated with the lack of immediate feedback from experimenters. Experimenters attempted to mitigate this concern by sending some feedback, positive or negative, for most captures. An unexpected side benefit of Momento seen in this study was that parents were pleased that their children could contribute to a study without being followed or watched by a stranger.

Experimenter experience

Momento facilitated many aspects of the experimental lifecycle. The experimenters were able to configure the study rapidly and run pilots using their own devices. They were then able to use the same settings (with the exception of the participants' mobile phone information) for the actual study. The experimenters then used the interface to monitor events during the study – in several cases they diagnosed problems with participants' mobile phones by noticing problems with captured data (*e.g.*, only text from a phone that supported MMS, or text added onto messages by the carrier network). As mentioned, experimenters also used the interface to provide feedback to participants, most commonly acknowledgement and encouragement.

Experimenters wanted to be able to use Momento during elicitation interviews. Specifically, they wanted to show individual participants their own timeline of images and to scroll through their events in the detail view. While this was already supported, it raised privacy concerns.

Experimenters also wanted to allow participants to annotate media captures from a web page, something that also arose in the Scribe4Me study. Also, we observed that some experimenters had difficulty sifting through hundreds of captures represented on some participants' timelines. Finally, experimenters easily imported Momento's data files into other programs to perform quantitative analysis.

Implications for design

This was the first extended use of Momento for diary studies, and as such it resulted in several significant additions. We expanded privacy support by augmenting the timeline to allow experimenters to switch between views that included all participants versus only one participant. We also added the ability to scroll through incoming message details per-participant while keeping experimenter notes private, and added a feature to print out physical copies of messages (including text and pictures) to support elicitation interviews at which a computer was not available. Lastly, we added an interactive table to the desktop platform listing all incoming messages. Experimenters found the list a more useful way of navigating incoming messages than the timeline for periods of high response rates (more than 20 per participant-hour).

6.3.5 PhotoSketch: Supporting informal classroom learning

Another group of external researchers are using the current version of Momento to conduct one-day workshops exploring a mobile learning application (PhotoSketch) for children in fifth grade (ages 10-11) at an elementary school. The researchers have thus far conducted two workshops, involving 11 and 13 participants and three and four experimenters, respectively. Our primary goal for this experiment was to observe how easily Momento could be extended to contexts for which it was not initially designed. We interviewed experimenters to understand the process of building a prototype application that integrated into a classroom scenario using Momento. We gathered feedback from the teachers as well.

Setup

The experimenters designed the workshop as a game for students, held at the students' schools. The objective of the game was for students to become more aware of physical mechanisms around them. The students were to move around the schoolyard, photograph objects with moving parts, and annotate the photographed objects to show which parts moved and how they moved. Afterwards, students were to share their thoughts regarding the photographs and sketches that they had created.

The experimenters configured the mobile client on a Pocket PC device to capture a photo and annotation. The experimenters also brought a laptop to the schools, primarily to monitor captured events in real-time (with the help of the desktop platform). The experimenters also ran a Momento server on the laptop and configured the laptop for peer-to-peer wireless networking. In this way, the mobile devices could connect directly to the server running on the laptop without depending on an external network (there was none at the schools).

To run the workshops, experimenters divided the participants into small groups and showed each group how to use the mobile application. They then let the students take and annotate photos. During this phase, the experimenters and the students' teacher monitored the groups to help them recover from mistakes (most commonly, accidentally hiding the application). Afterwards, the experimenters used the Momento desktop platform to review captures with the students in a classroom setting.

Participant experience

Although this experiment required participants to use a new device, it avoided some issues such as carrying an extra device around over days by being constrained to a workshop setting. In follow up interviews, experimenters reported that participants had little to say about the mobile application itself, but it was clear that with only a brief training session they were easily able to use the application to take and annotate photos. The only problems reported were that occasionally one device launched a different application, and participants sometimes clicked on the operating system button that hid the main screen. In both cases participants recovered quickly and were minimally distracted from the main task. The experimenters also noted that participants universally liked using the application.

The teachers' experiences were also largely positive. Based on her experience with PhotoSketch, one teacher is integrating a series of similar workshops into her curriculum for the coming school year.

Experimenter experience

Experimenters felt that using Momento was a large improvement over their fallback approach, having students manually draw both the object and the motion of the object. Because of the work involved in coordinating with schools and teachers and developing and iterating the overall workshop design, experimenters had little time to develop a mobile prototype, but were able to do so with Momento in minutes. One experimenter said that the use of a mobile application "simply wouldn't have happened" without Momento.

Experimenters wanted slightly more control configuring the mobile client than was supported at the time. For example, they asked us to add support for a color scheme and lengthier text description.

Also, for this application experimenters needed to configure the desktop platform, server, and client to work on an ad hoc network. To support this, no adjustments were necessary to the desktop platform nor mobile client. However, the server configuration had involved some complicated manual processes that the experimenters initially found too difficult to complete.

Experimenters did report that in some cases some images did not transmit properly from the mobile device to the desktop platform. These issues may have been due to interference in the *ad hoc* network the experimenters had configured. However, experimenters were able to synchronize the mobile clients with the laptop to recover all images.

Implications for design

This study validated that Momento can be used in some evaluation contexts for which it was not initially designed. Our principle discovery was the need to streamline server configuration to the point that the experimenters needed only to run four commands to configure the core system and the study.

6.3.6 Discussion

Overall, our studies have shown that Momento can be a powerful tool for ubicomp experimenters and a usable tool for participants. Although the studies just described provided overall validation for our concept, they also provided valuable feedback about the features and structure of Momento, much of which was integrated in to the current version.

Some of the important features of Momento that were influenced or identified during iteration include:

- Added a notion of places, groups, and rules to the desktop platform to support social-mobile applications (Pilot)
- Built mobile application to support more sophisticated applications (Pilot)
- Streamlined response process (Scribe4Me)
- Removal of modal notifications (Scribe4Me)
- Increased support for piggybacking on existing devices (AwarenessBoard)
- Addition of support for time and place (AwarenessBoard)
- Expanded and more sophisticated networking support (AwarenessBoard)
- Better privacy support when viewing data on the desktop platform (Informal Diary)
- Better support for sifting through hundreds of captures, an opposite problem to data sparsity (Informal Diary)
- Privacy-sensitive web access to participant data (Informal Diary)

Participants generally found Momento usable. Most of the problems encountered by participants involved usability issues arising from the platforms and networks on which the mobile client was deployed. These drawback stem from our reliance on everyday infrastructure, and the AwarenessBoard and Scribe4Me study showed that these problems can affect adoption, retention and data quality and quantity.

Our results indicate that experimenters are able to configure Momento rapidly for a variety of different types of experiments. Also, our results indicate that, in general, experimenters want as much automation as possible. Custom programming of ubicomp applications is a high barrier for many experimenters considering running a situated study. Our rules system and text file configuration help to reduce the need for custom programming. Also, it can be difficult for experimenters to interpret and respond to events while also attempting to complete other work. This issue can make it difficult to run experiments for long periods. However, as the InformalLearning and Scribe4Me studies showed Momento does provide some support for distributing tasks across mobile experimenters, and while this approach was hampered by infrastructure delays in our studies, improving networks may make it more viable. Also, as the PhotoSketch study showed, Momento can be useful even in the absence of any external connectivity infrastructure whatsoever.

Experimenters used Momento's logging system both for real-time and *post-hoc* analysis. Especially in the AwarenessBoard study, experimenters were able to move smoothly from the prototyping to the analysis stage of iteration. In other studies, quantitative data from Momento and qualitative data from interviews tended to center around user capture events – future work could explore explicitly bridging the two.

We have found that some issues stand in the way of large scale deployment, including problems unique to certain networks and devices as well as users less familiar with mobile devices. Thus, researchers using Momento should expect to pilot their studies and provide support and training to users. But, again, our findings indicate that overall Momento makes situated, rapid iteration newly available to a wide variety of applications.

Chapter 7

Conclusion

In this thesis, we have illustrated that sensing and scale present roadblocks to ubicomp iteration and have offered some guidelines and tools for overcoming these roadblocks. Specifically, our interviews with developers in three subfields of ubicomp – mobile applications, peripheral displays, and tangible user interfaces – combined with an extensive literature review showed that achieving ecologically valid iterative design for ubicomp is a struggle. From this body of work, we derived five central challenges for situated evaluation of ubicomp – ambiguity and error, sparse data, critical mass, unobtrusiveness, and tool support for realistic environments. We then described how the diary study method can be modified to overcome some of these challenges. Specifically, we showed that media-based diary studies can help address data sparsity while remaining relatively unobtrusive.

From our literature review, interviews, and formative work with the diary study method, we derived a set of requirements to support realistic ubicomp experiments and built a tool based on these requirements, Momento. Momento provides a desktop platform that connects experimenters with participants in the field, and includes a simple, configurable application extending the capabilities of mobile devices to support participant data collection. We designed Momento iteratively to meet the needs of participants and experimenters in four separate studies that involved a variety of tools and evaluation methods.

In the next section, we describe issues that were either beyond the scope of the thesis or were only discovered retrospectively.

7.1 Future work

One of the fundamental challenges not addressed by this thesis is supporting synchronous interaction and streaming. Much support has focused on either asynchronous access and annotation or synchronous video streaming. However, these approaches have not completely integrated some of the fundamental limitations but also richness of mobile devices. In particular, 1) users accessing content on mobile devices are often dividing their attention between multiple tasks, 2) devices have limited output capability (screen real estate, *etc.*), and 3) users may need to rapidly switch interaction styles to compensate for changing context. To compensate for these issues, synchronous support systems should 1) support discrete interaction as well as awareness of participants' state and backchannel communication to "catch-up" with missed material 2) filter content (*e.g.*, interactively find the more critical information in a distributed meeting) and 3) provide manual or automatic controls to change the view.

However, mobile use also has the advantage of acquiring data from field settings that can potentially augment some types of synchronous interactions. For example, in a collaborative activity, it may be useful to gather recordings, images, and video from the field to augment conversation around a particular topic (content available online can only go so far). One of the challenges here is to build a system that supports fluidly moving between synchronous and asynchronous use.

There also remain holes in ubicomp evaluation methodologies. First, while this thesis makes the case that realism is important in evaluating ubicomp applications, there are some aspects of realistic deployments that do not necessarily add to the iterative development of a system. More work could be done to determine how to structure field deployments. For example, while it is important to test a system's robustness to network and device variations, it is usually not useful to ensure that the network and device being tested have exactly the same characteristics as those that are the most popular currently. In the Momento system, we encountered this problem numerous times with regards to cellular operators. There were few additional benefits derived from developing a system that relied on only cellular devices and connectivity as compared to a more open platform that could make use of a variety of networks (*e.g.*, 802.11). Furthermore, as the Photosketch study showed, it can be useful to have complete control over the network. In general, more research should be done to discern the optimum balance between realistic and contrived field studies.

Finally, ubicomp *research* is in danger of becoming smart-phone-centric. While tabs are one of the elements of the core technological vision of ubicomp (tabs, pads, boards), as has been discussed earlier this thesis takes the position that ubicomp is fundamentally about supporting more aspects of activities and doing so in a way that compliments current practice. While there exists support for rapidly prototyping ubicomp technologies that do not necessarily involve mobile phones, more work needs to be done to develop evaluation methodologies for situations in which mobile phones are not appropriate or available at all (sports, automobiles, deep sea sensor networks, *etc.*). However, as ubicomp envelopes ever more activities, work attempting to encompass its whole will become untenable and too generic to be useful. Thus, research in this area will be most useful if it concentrates on guidelines derived from situated studies of realistic deployments, rather than on technical solutions *per se*, and if it tailors those guidelines to specific issues (massive sensor deployments, non-display environments, multi-display environments, *etc.*) rather than broad claims about all of ubicomp.

Bibliography

[1] Gregory D. Abowd. Classroom 2000: An experiment with the instrumentation of a living educational environment. *IBM Systems Journal*, 38(4):508–530, 1999.

[2] Gregory D. Abowd. Software engineering issues for ubiquitous computing. In *Proceedings of the International Conference on Software Engineering (ICSE)*, pages 75–84, 1999.

[3] Gregory D. Abowd, Christopher G. Atkeson, Aaron F. Bobick, Irfan A. Essa, Blair MacIntyre, Elizabeth D. Mynatt, and Thad E. Starner. Living Laboratories: The Future Computing Environments Group at the Georgia Institute of Technology. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI)*, pages 215–216, 2000.

[4] Gregory D. Abowd, Gillian R. Hayes, Giovanni Iachello, Julie A. Kientz, Shwetak N. Patel, and Molly M. Stevens. Prototypes and paratypes: Designing mobile and ubiquitous computing applications. *IEEE Pervasive Computing*, 4(4):67–73, 2005.

[5] Gregory D. Abowd and Elizabeth D. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(1):29–58, 2000.

[6] David Akers. Cinch: A cooperatively designed marking interface for 3D pathway selection. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, pages 33–42, 2006.

[7] J. Allanson, T. Rodden, and J. Mariani. A toolkit for exploring electrophysiological human-computer interaction. In *Proceedings of IFIP INTERACT: Human-Computer Interaction*, pages 231–237, 1999.

[8] Mike Ananny, Kathleen Biddick, and Carol Strohecker. Constructing public discourse with ethnographic/sms "texts?". In *Proceedings of the International Conference on Human Computer Interaction with Mobile Devices and Services (Mobile HCI)*, pages 368–373, 2003.

[9] R. J. Anderson. Representations and requirements: The value of ethnography in system design. *Human-Computer Interaction*, 9:151–182, 1994.

[10] Rafael Ballagas, Meredith Ringel, Maureen Stone, and Jan Borchers. iStuff: A physical user interface toolkit for ubiquitous computing environments. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI)*, pages 537–544, 2003.

[11] Lisa Feldman Barrett and Daniel J. Barrett. An introduction to computerized experience sampling in psychology. *Social Science Computer Review*, 19(2):175–185, 2001.

[12] L. W. Barsalou. The content and organization of autobiographical memories. In U. Neisser and E. Winograd, editors, *Remembering reconsidered: Ecological and traditional approaches to the study of memory*, pages 193–243. Cambridge University Press, 1988.

[13] Jennifer Beaudin, Stephen Intille, and Emmanuel Munguia Tapia. Lessons learned using ubiquitous sensors for data collection in real homes. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI)*, pages 1359–1362, 2004.

[14] Victoria Bellotti, Maribeth Back, W. Keith Edwards, Rebecca E. Grinter, D. Austin Henderson Jr., and Cristina Videira Lopes. Making sense of sensing systems: Five questions for designers and researchers. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 415–422, 2002.

[15] Steve Benford, Andy Crabtree, Martin Flintham, Adam Drozd, Rob Anastasi, Mark Paxton, Nick Tandavanitj, Matt Adams, and Ju Row-Farr. Can you see me now? *ACM Transactions on Computer-Human Interaction*, 13(1):100–133, 2006.

[16] Steve Benford, Will Seager, Martin Flintham, Rob Anastasi, Duncan Rowland, Jan Humble, Danae Stanton, John Bowers, Nick Tandavanitj, Matt Adams, Ju Row-Farr, Amanda Oldroyd, and Jon Sutton. The error of our ways: The experience of self-reported position in a location-based game. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp)*, pages 70–87, 2004.

[17] Betiol, A. H., Cybis, and W. de Abreu. Usability testing of mobile devices: A comparison of three approaches. In *Proceedings of IFIP INTERACT: Human-Computer Interaction*, pages 470–481, 2005.

[18] Hugh Beyer and Karen Holtzblatt. *Contextual Design: Defining Customer-centered Systems*. Series in Interactive Technologies. Morgan Kaufmann, San Francisco, CA, 1998.

[19] Joel Brandt and Scott R. Klemmer. Lash-ups: A toolkit for location-aware mash-ups. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, 2006. Poster.

[20] Barry A. T. Brown, Abigail J. Sellen, and Kenton P. O'Hara. A diary study of information capture in working life. In *Conference on Human Factors in Computing Systems (CHI)*, pages 438–445, 2000.

[21] Marion Buchenau and Jane Fulton Suri. Experience prototyping. In *Proceedings of the Symposium on Designing Interactive Systems (DIS)*, pages 424–433, 2000.

[22] J.J. Cadiz, G. Venolia, G. Jancke, and A. Gupta. Designing and deploying an information awareness interface. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW)*, pages 314–323, 2002.

[23] Stefan Carmien, Rogerio DePaula, Andrew Gorman, and Anja Kintsch. Increasing workplace independence for people with cognitive disabilities by leveraging distributed cognition among caregivers and clients. In *Proceedings of the Conference on Supporting Group Work (GROUP)*, pages 95–104, 2003.

[24] S. Carter, J. Mankoff, S. Klemmer, and T. Matthews. Exiting the cleanroom: On ecological validity and ubiquitous computing. *Human-Computer Interaction Journal*, 2007. To appear.

[25] Scott Carter and Jennifer Mankoff. Prototypes in the wild: Lessons learned from evaluating three ubicomp systems. *IEEE Pervasive Computing*, 4:51–57, 2005.

[26] Scott Carter and Jennifer Mankoff. When participants do the capturing: The role of media in diary studies. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 899–908, 2005.

[27] Scott Carter, Jennifer Mankoff, and Patrick Goddi. Building connections among loosely coupled groups: Hebb's rule at work. *Journal of Computer Supported Cooperative Work*, 13:305–327, 2004.

[28] Scott Carter, Jennifer Mankoff, and Jeffrey Heer. Support for situated ubicomp experimentation. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 2007. To appear.

[29] Corey D. Chandler, Gloria Lo, and Anoop K. Sinha. Multimodal theater: Extending low fidelity paper prototyping to multimodal applications. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI)*, Student Posters, pages 874–875, 2002.

[30] Ana Ramirez Chang and Marc Davis. Designing systems that direct human action. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI)*, pages 1260–1263, 2005.

[31] Elizabeth F. Churchill, Les Nelson, Laurent Denoue, Jonathan Helfman, and Paul Murphy. Sharing multimedia content with interactive public displays: A case study. In *Proceedings of the Symposium on Designing Interactive Systems (DIS)*, pages 7–16, 2004.

[32] Marisol Clark-Ibáñez. Framing the social world with photo-elicitation interviews. *American Behavioral Scientist*, 47(12):1507–1527, 2004.

[33] Martin Colbert. A diary study of rendezvousing: Implications for position-aware computing and communications for the general public. In *Proceedings of the Conference on Supporting Group Work (GROUP)*, pages 15–23, 2001.

[34] S. Consolvo and M. Walker. Using the experience sampling method to evaluate ubicomp applications. *IEEE Pervasive Computing*, 2(2):24–31, 2003.

[35] Sunny Consolvo, Peter Roessler, and Brett E. Shelton. The CareNet display: Lessons learned from an in home evaluation of an ambient display. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp)*, pages 1–17, 2004.

[36] Sunny Consolvo, Ian E. Smith, Tara Matthews, Anthony LaMarca, Jason Tabert, and Pauline Powledge. Location disclosure to social relations: Why, when, & what people want to share. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 81–90, 2005.

[37] Andy Crabtree, Steve Benford, Chris Greenhalgh, Paul Tennent, Matthew Chalmers, and Barry Brown. Supporting ethnographic studies of ubiquitous computing in the wild. In *Proceedings of the Symposium on Designing Interactive Systems (DIS)*, pages 60–69, 2006.

[38] Cybertracker. http://www.cybertracker.co.za/.

[39] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. Wizard of Oz studies – why and how. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, pages 193–200, 1993.

[40] Nigel Davies, James Landay, Scott Hudson, and Albrecht Schmidt. Rapid prototyping for ubiquitous computing. *IEEE Pervasive Computing*, 4(4):15–17, 2005.

[41] Françoise Détienne. *Software Design-Cognitive Aspects*. Practitioner series. Springer Verlag, London, 2001.

[42] Anind K. Dey. *Providing Architectural Support for Building Context-Aware Applications*. PhD thesis, Georgia Institute of Technology, 2000.

[43] Anind K. Dey, Gregory D. Abowd, and Daniel Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2/4):97–166, 2001.

[44] Anind K. Dey, R. Hamid, Christopher Beckmann, Ian Li, and Daniel Hsu. *a CAPpella*: Programming by demonstration of context-aware applications. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 33–40, 2004.

[45] Paul Dourish. *Where the action is: The foundations of embodied interaction.* MIT Press, Cambridge, MA, 2001.

[46] Steven Dow, Jaemin Lee, Christopher Oezbek, Blair MacIntyre, Jay David Bolter, and Maribeth Gandy. Wizard of Oz interfaces for mixed reality applications. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI)*, pages 1339–1342, 2005.

[47] W. Keith Edwards, Victoria Bellotti, Anind K. Dey, and Mark W. Newman. The challenges of user-centered design and evaluation for infrastructure. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 297–304, 2003.

[48] W. Keith Edwards and Rebecca E. Grinter. At home with ubiquitous computing: Seven challenges. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp)*, pages 256–272, 2001.

[49] Margery Eldridge, Michael Lamming, and Mike Flynn. Does a video diary help recall? In *Proceedings of People and Computers VII (HCI)*, pages 257–269, 1992.

[50] George W. Fitzmaurice, Hiroshi Ishii, and William Buxton. Bricks: Laying the foundations for graspable user interfaces. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 442–449, 1995.

[51] Margaret Fleck, Marcos Frid, Tim Kindberg, Eamonn O'Brien-Strain, Rakhi Rajani, and Mirjana Spasojevic. Rememberer: A tool for capturing museum visits. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp)*, pages 48–55, 2002.

[52] James Fogarty, Scott Hudson, C. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J. Lee, and J. Yang. Predicting human interruptibility with sensors. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(1):119–146, 2005.

[53] Jon Froehlich, Mike Y. Chen, Ian E. Smith, and Fred Potter. Voting with your feet: An investigative study of the relationship between place visit behavior and preference. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp)*, pages 333–350, 2006.

[54] Bill Gaver, Tony Dunne, and Elena Pacenti. Design: Cultural probes. *Interactions*, 6(1):21–29, 1999.

[55] V. M. Gonzalez and G. Mark. 'Constant, constant, multi-tasking craziness': Managing multiple working spheres. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 113–120, 2004.

[56] Saul Greenberg and Chester Fitchett. Phidgets: Easy development of physical interfaces through physical widgets. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, pages 209–218, 2001.

[57] Robert Grimm, Janet Davis, Eric Lemar, Adam Macbeth, Steven Swanson, Thomas Anderson, Brian Bershad, Gaetano Borriello, Steven Gribble, and David Wetherall. System support for pervasive applications. *ACM Transactions on Computer Systems (TOCS)*, 22(4):421–486, 2004.

[58] Rebecca Grinter and Margery Eldridge. Wan2tlk?: Everyday text messaging. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 441–448, 2003.

[59] Rebecca E. Grinter and Margery Eldridge. y do tngrs luv 2 txt msg? In *European Conference on Computer-Supported Cooperative Work (ECSCW)*, pages 219–238, 2001.

[60] K. Grønbæk, M. Kyng, and P. Mogensen. CSCW challenges in large-scale technical projects–a case study. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW)*, pages 338–345, 1992.

[61] Jonathan Grudin. Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM*, 37(1):92–105, 1994.

[62] Jonna Häkkilä and Jani Mäntyjärvi. Combining location-aware mobile phone applications and multimedia messaging. *Mobile Multimedia*, 1(1):18–32, 2005.

[63] John Halloran, Eva Hornecker, Geraldine Fitzpatrick, Mark Weal, David Millard, Danius Michaelides, Don Cruickshank, and David De Roure. Unfolding understandings: Co-designing ubicomp in situ, over time. In *Proceedings of the Symposium on Designing Interactive Systems (DIS)*, pages 109–118, 2006.

[64] M. Hammersley and Paul Atkinson. *Ethnography: Principles in Practice*. Routledge, London, 1995.

[65] Björn Hartmann, Scott R. Klemmer, Michael Bernstein, Leith Abdulla, Brandon Burr, Avi Robinson-Mosher, and Jennifer Gee. Reflective physical prototyping through integrated design, test, and analysis. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, pages 299–308, 2006.

[66] J. Heer, S. K. Card, and J. A. Landay. prefuse: A toolkit for interactive information visualization. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 421–430, 2005.

[67] T. Hemmings, A. Crabtree, T. Rodden, K. Clarke, and M. Rouncefield. Probing the probes: Domestic probes and the design process. In *Proceedings of the European Conference on Cognitive Ergonomics*, pages 8–11, 2002.

[68] J.D. Herbsleb, D.L Atkins, D.G. Boyer, M. Handel, and T.A. Finholt. Introducing instant messaging and chat in the workplace. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 171–178, 2002.

[69] Ralph D. Hill. Supporting concurrency, communication and synchronization in human-computer interaction — the Sassafras UIMS. *ACM Transactions on Graphics (TOG)*, 5(3):179–210, 1986.

[70] Wai-Ling Ho-Ching, Jennifer Mankoff, and James Landay. Can you see what I hear? The design and evaluation of a peripheral sound display for the deaf. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 161–168, 2003.

[71] R. M. Hogarth. Is confidence in decisions related to feedback? In Klaus Fiedler and Peter Juslin, editors, *In the beginning there is a sample: Information sampling as a key to understand adaptive cognition*. Cambridge University Press, 2004.

[72] Lars Erik Holmquist. Prototyping: Generating ideas or cargo cult designs? *Interactions of the ACM*, 12(2):48–54, 2005.

[73] Karen Holtzblatt, Jessamyn Burns Wendell, and Shelley Wood. *Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design*. Morgan Kaufmann, San Francisco, CA, 2005.

[74] Jason I. Hong and James A. Landay. An infrastructure approach to context-aware computing. *Human-Computer Interaction*, 16(2/4):287–303, 2001.

[75] Heather Horst and Daniel Miller. From kinship to link-up: Cell phones and social networking in Jamaica. *Current Anthropology*, 46(5):755–778, 2005.

[76] S. Houde and C. Hill. What do prototypes prototype? In M. Helander, T. E. Landauer, and P. Prabhu, editors, *Handbook of Human-Computer Interaction*. Elsevier Science, Amsterdam, second edition, 1997.

[77] S. Hudson, J. Fogarty, C. Atkeson, J. Forlizzi, S. Kielser, J. Lee, and J. Yang. Predicting human interruptibility with sensors: A Wizard of Oz feasibility study. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 257–264, 2003.

[78] S. Hulkko, T. Mattelmäki, K. Virtanen, and T. Keinonen. Mobile probes. In *Proceedings of the Nordic Conference on Human-Computer Interaction (NordiCHI)*, pages 43–51, 2004.

[79] Hilary Hutchinson, Wendy Mackay, Bosse Westerlund, Benjamin B. Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stephane Conversy, Helen Evans, Heiko Hansen, Nicolas Roussel, and Björn Eiderback. Technology probes: Inspiring design for and with families. In *Conference on Human Factors in Computing Systems (CHI)*, pages 17–24, 2003.

[80] Giovanni Iachello, Khai Truong, Gregory Abowd, Gillian Hayes, and Molly Stevens. Event-contingent experience sampling to evaluate ubicomp technology in the real world. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 1009–1018, 2006.

[81] Intel Research Laboratory, Seattle. iESP. http://seattleweb.intel-research.net/projects/esm/iESP.html.

[82] S. S. Intille, K. Larson, E. Munguia Tapia, J.S. Beaudin, P. Kaushik, J. Nawyn, and R. Rockinson. Using a live-in laboratory for ubiquitous computing research. In *Proceedings of the International Conference on Pervasive Computing (Pervasive)*, pages 349–365, 2006.

[83] Stephen Intille, Charles Kukla, and Xiaoyi Ma. Eliciting user preferences using image-based experience sampling and reflection. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 738–739, 2002.

[84] Stephen S. Intille, Kent Larson, J. S. Beaudin, J. Nawyn, E. Munguia Tapia, and P. Kaushik. A living laboratory for the design and evaluation of ubiquitous computing technologies. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI)*, pages 1941–1944, 2005.

[85] Stephen S. Intille, Emmanuel Munguia Tapia, John Rondoni, Jennifer Beaudin, Chuck Kukla, Sitij Agarwal, Ling Bao, and Kent Larson. Tools for studying behavior and technology in natural settings. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp)*, pages 157–174, 2003.

[86] Jacob and Robert J. K. What you look at is what you get: Eye movement-based interaction techniques. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, Eye, Voice and Touch, pages 11–18, 1990.

[87] Brad Johanson, Terry Winograd, and Armando Fox. Interactive workspaces. *IEEE Computer*, 36(4):99–101, 2003.

[88] F. Thomas Juster. Conceptual and methodological issues involved in the measurement of time use. In F. Thomas Juster and Frank P. Stafford, editors, *Time, Goods, and Well-Being*, pages 19–32. Institute for Social Research, The University of Michigan, 1985.

[89] Anne Kaikkonen, Aki Kekäläinen, Mihael Cankar, Titti Kallio, and Anu Kankainen. Usability testing of mobile applications: A comparison between laboratory and field testing. *Journal of Usability Studies*, 1(1):4–17, 2005.

[90] John F. Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41, 1984.

[91] Tom Kelley. *The Art of Innovation*. Doubleday, New York, NY, 2001.

[92] Cory D. Kidd, Robert Orr, Gregory D. Abowd, Christopher G. Atkeson, Irfan A. Essa, Blair MacIntyre, Elizabeth D. Mynatt, Thad Starner, and Wendy Newstetter. The aware home: A living laboratory for ubiquitous computing research. In *Proceedings of Cooperative Buildings, Integrating Information, Organization, and Architecture, Second International Workshop (CoBuild)*, pages 191–198, 1999.

[93] T. Kindberg, M. Spasojevic, and A. Sellen. R. Fleck. The ubiquitous camera: An in-depth study of camera phone use. *IEEE Pervasive Computing*, 4(2):42–50, 2005.

[94] J. Kjeldskov, C. Graham, S. Pedell, F. Vetere, S. Howard, S. Balbo, and J. Davies. Evaluating the usability of a mobile guide: The influence of location, participants and resources. *Behaviour and Information Technology*, 24(1):51–65, 2005.

[95] Jesper Kjeldskov and Connor Graham. A review of mobile HCI research methods. In *Proceedings of the International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 317–335, 2003.

[96] Jesper Kjeldskov, Mikael B. Skov, Benedikte S. Als, and Rune T. Høegh. Is it worth the hassle? Exploring the added value of evaluating the usability of context-aware mobile systems in the field. In *Proceedings of the International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 61–73, 2004.

[97] Jesper Kjeldskov and Jan Stage. New techniques for usability evaluation of mobile systems. *International Journal of Human-Computer Studies*, 60(5-6):599–620, 2004.

[98] Scott R. Klemmer. *Tangible User Interfaces: Tools and Techniques*. PhD thesis, University of California, Berkeley, 2004.

[99] Scott R. Klemmer, Björn Hartmann, and Leila Takayama. How bodies matter: Five themes for interaction design. In *Proceedings of the Symposium on Designing Interactive Systems (DIS)*, pages 140–149, 2006.

[100] Scott R. Klemmer, Jack Li, James Lin, and James Landay. Papier-mâché: Toolkit support for tangible input. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 399–406, 2004.

[101] Scott R. Klemmer, Mark W. Newman, Ryan Farrell, Mark Bilezikjian, and James A. Landay. The designers' outpost: A tangible interface for collaborative web site. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, pages 1–10, 2001.

[102] Scott R. Klemmer, Anoop K. Sinha, Jack Chen, James A. Landay, Nadeem Aboobaker, and Annie Wang. SUEDE: A Wizard of Oz prototyping tool for

speech user interfaces. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, pages 1–10, 2000.

[103] M. Lamming and M. Flynn. Forget-me-not: Intimate computing in support of human memory. In *Proceedings of FRIEND21 Symposium on Next Generation Human Interfaces*, 1994.

[104] James A. Landay. *Interactive Sketching for the Early Stages of User Interface Design*. PhD thesis, Carnegie Mellon University, 1996.

[105] Brian Lee, Heidy Maldonado, Scott Klemmer, Isabelle Kim, and Paz Hilfinger-Pardo. Longitudinal studies of augmented notebook usage informing the design of sharing mechanisms. Technical Report CSTR 2006-11, Stanford University, 2006.

[106] Francis Li. Mobile processing. http://mobile.processing.org/, 2006.

[107] Yang Li, Jason I. Hong, and James A. Landay. Topiary: A tool for prototyping location-enhanced applications. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, pages 217–226, 2004.

[108] L. Liu and P. Khooshabeh. Paper or interactive? A study of prototyping techniques for ubiquitous computing environments. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 1030–1031, 2003.

[109] Allan Christian Long, Jr., James A. Landay, and Lawrence A. Rowe. Implications for a gesture design tool. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 40–47, 1999.

[110] Blair MacIntyre, Maribeth Gandy, Steven Dow, and Jay David Bolter. DART: A toolkit for rapid design exploration of augmented reality experiences. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, pages 197–206, 2004.

[111] Wendy E. Mackay. *Users And Customizable Software: A Co-Adaptive Phenomenon*. PhD thesis, Sloan School of Management, Massachusetts Institute of Technology, 1990.

[112] Wendy E. Mackay. Triangulation within and across HCI disciplines. *Human-Computer Interaction*, 13(3):310–315, 1998.

[113] Wendy E. Mackay, Anne-Laure Fayard, Laurent Frobert, and Lionel Médini. Reinventing the familiar: Exploring an augmented reality design space for air traffic control. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 558–565, 1998.

[114] Jennifer Mankoff, Anind K. Dey, Gary Hsieh, Julie Kientz, Morgan Ames, and Scott Lederer. Heuristic evaluation of ambient displays. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 169–176, 2003.

[115] Jennifer Mankoff, Scott E. Hudson, and Gregory D. Abowd. Interaction techniques for ambiguity resolution in recognition-based interfaces. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, pages 11–20, 2000.

[116] Jennifer Mankoff and Bill Schilit. Supporting knowledge workers beyond the desktop with PALPlates. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI)*, pages 550–551, 1997.

[117] T. Matthews, A. Dey, J. Mankoff, S. Carter, and T. Rattenbury. A toolkit for managing user attention in peripheral displays. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, pages 247–256, 2004.

[118] T. Matthews, J. Fong, F.W. Ho-Ching, and J. Mankoff. Evaluating non-speech sound visualizations for the deaf. *Behaviour and Information Technology*, 25(4):333–351, 2006.

[119] Tara Matthews. Interviewing peripheral display designers and developers. Technical Report EECS-2005-19, University of California, Berkeley, 2005.

[120] Tara Matthews, Scott Carter, Janette Fong, Carol Pai, and Jennifer Mankoff. Scribe4me: Evaluating a mobile sound translation tool for the deaf. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp)*, pages 159–176, 2006.

[121] Tara Matthews, Mary Czerwinski, George Robertson, and Desney Tan. Clipping lists and change borders: Improving multitasking efficiency with peripheral information design. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 989–998, 2006.

[122] Tara Matthews, Tye Rattenbury, and Scott Carter. Defining, designing, and evaluating peripheral displays: An analysis using activity theory. *Human-Computer Interaction*, 22(1/2), 2007.

[123] David Maulsby, Saul Greenberg, and Richard Mander. Prototyping an intelligent agent through Wizard of Oz. In *Proceedings of the Conference on Human Factors in Computing Systems (INTERCHI)*, pages 277–284, 1993.

[124] D. Scott McCrickard, C. M. Chewar, Jacob P. Somervell, and Ali Ndiwalana. A model for notification systems evaluation – assessing user goals for multitasking activity. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 10(4):312–338, 2003.

[125] McGee, David R., Cohen, Philip R., Wesson, R. Matthews, and Sheilah Horman. Comparing paper and tangible, multimodal tools. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 407–414, 2002.

[126] Joseph E. McGrath. Methodology matters: Doing research in the behavioral and social sciences. In R. M. Baecker, J. Grudin, and W. A. S. Buxton, editors, *Readings in Human-Computer Interaction: Toward the Year 2000*, pages 152–169. Morgan Kaufman, San Francisco, CA, 1995.

[127] Thomas P. Moran, Patrick Chiu, Steve Harrison, Gordon Kurtenbach, Scott Minneman, and William van Melle. Evolutionary Engagement in an ongoing collaborative work process: A case study. In *Computer Supported Cooperative Work (CSCW)*, pages 150–159, 1996.

[128] Thomas P. Moran, Patrick Chiu, and William van Melle. Pen-based interaction techniques for organizing material on an electronic whiteboard. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, pages 45–54, 1997.

[129] Alistair Morrison, Paul Tennent, and Matthew Chalmers. Coordinated visualisation of video and system log data. In *Proceedings of the International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV)*, pages 91–102, 2006.

[130] E.D. Mynatt, J. Rowan, S. Craighill, and A. Jacobs. Digital family portraits: Providing peace of mind for extended family members. In *Proceedings of Conference on Human Factors in Computing Systems (CHI)*, pages 333–340, 2001.

[131] Mark W. Newman, James Lin, Jason I. Hong, and James A. Landay. DENIM: An informal web site design tool inspired by observations of practice. *Human-Computer Interaction*, 18(3):259–324, 2003.

[132] J. Nielsen. Usability engineering at a discount. In G. Salvendy and M.J. Smith, editors, *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, pages 394–401. Elsevier Science, Amsterdam, 1989.

[133] Nokia. Python for series 60. http://www.forum.nokia.com/python/.

[134] Kenton O'Hara and Mark Perry. Shopping anytime anywhere. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 345–346, 2001.

[135] D. Okabe, K. Anderson, M. Ito, and S. Mainwairing. Location-based moblogging as method: New views into the use and practice of personal, social, and mobile technologies. In *Online Proceedings of Seeing, Understanding, Learning in the Mobile Age*, 2005.

[136] Daisuke Okabe and Mizuko Ito. Everyday contexts of camera phone use: Steps toward technosocial ethnographic frameworks. In J. Höflich and M. Harmann, editors, *Mobile Communication in Everyday Life: An Ethnographic View*, pages 79–102. Frank and Timme, Berlin, 2006.

[137] Dan R. Olsen, Trent Taufer, and Jerry Alan Fails. ScreenCrayons: Annotating anything. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, pages 165–174, 2004.

[138] Gary M. Olson and Judith S. Olson. Distance matters. *Human-Computer Interaction*, 15(2/3):139–178, 2000.

[139] J. W. Orlikowski. The duality of technology: Rethinking the concept of technology in organizations. *Organization Science*, 3:398–427, 1992.

[140] Antti Oulasvirta, Sakari Tamminen, Virpi Roto, and Jaana Kuorelahti. Interaction in 4-second bursts: The fragmented nature of attentional resources in mobile HCI. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 919–928, 2005.

[141] Sharon Oviatt, Phil Cohen, Lizhong Wu, Lisbeth Duncan, Bernhard Suhm, Josh Bers, Thomas Holzman, Terry Winograd, James Landay, Jim Larson, and David Ferro. Designing the user interface for multimodal speech and pen-based gesture applications: State-of-the-art systems and future research directions. *Human-Computer Interaction*, 15(4):263–322, 2000.

[142] Leysia Palen and Marilyn Salzman. Voice-mail diary studies for naturalistic data capture under mobile conditions. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW)*, pages 87–95, 2002.

[143] Leysia Palen, Marilyn C. Salzman, and Ed Youngs. Going wireless: Behavior & practice of new mobile phone users. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW)*, pages 201–210, 2000.

[144] Jason Pascoe, Nick Ryan, and David Morse. Using while moving: HCI issues in fieldwork environments. *ACM Transactions on Computer-Human Interaction*, 7(3):417–437, 2000.

[145] Eric Paulos and Elizabeth Goodman. The familiar stranger: Anxiety, comfort, and play in public places. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 223–230, 2004.

[146] Eric Paulos and Tom Jenkins. Urban probes: Encountering our emerging urban atmospheres. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 341–350, 2005.

[147] Marianne Graves Petersen, Kim Halskov Madsen, and Arne Kjaer. The usability of everyday technology: Emerging and fading opportunities. *ACM Transactions on Computer-Human Interaction (TOCHI))*, 9(2):74–105, 2002.

[148] N. Postman. *Technopoly*. Vintage, New York, NY, 1992.

[149] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen. ContextPhone - a prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing*, 4:51–59, 2005.

[150] Marc Rettig. Practical programmer: Prototyping for tiny fingers. *Communications of the ACM*, 37(4):21–27, 1994.

[151] John Rieman. The diary study: A workplace-oriented research tool to guide laboratory efforts. In *Proceedings of the Conference on Human Factors in Computing Systems (INTERCHI)*, pages 321–326, 1993.

[152] J. C. Rondini. Context-aware experience sampling for the design and study of ubiquitous technologies. Master's thesis, EECS Department, Massachusetts Institute of Technology, 2003.

[153] Jim Rowan and Elizabeth D. Mynatt. Digital family portrait field trial: Support for aging in place. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 521–530, 2005.

[154] Jeffrey Rubin. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. John Wiley and Sons, New York, NY, 1994.

[155] Åsa Rudström, Rickard Cöster, Kristina Höök, and Martin Svensson. Paper prototyping a social mobile service. In *Proceedings of the Workshop on Designing for Ubicomp in the Wild: Methods for Exploring the Design of Mobile and Ubiquitous Services (MUM)*, 2003.

[156] Alice Vera Sampson-Cordle. *Exploring the relationship between a small rural school in Northeast Geogia and its community: An image-based study using participant-produced photographs*. PhD thesis, University of Georgia, 2001.

[157] Jeffrey Samuels. Breaking the ethnographer's frames: Reflections on the use of photo elicitation in understanding sri lankan monastic culture. *American Behavioral Scientist*, 47(12):1528–1550, 2004.

[158] Albrecht Schmidt, Van de Velde, Walter, and Gerd Kortuem. Situated interaction in ubiquitous computing. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI)*, page 374, 2000.

[159] Donald A. Schön and John Bennett. Reflective conversation with materials. In Terry Winograd, editor, *Bringing Design to Software*. ACM Press, New York, NY, 1996.

[160] Abigail Sellen and Richard Harper. Paper as an analytic resource for the design of new technologies. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 319–326, 1997.

[161] Abigail Sellen and Richard H. R. Harper. *The Myth of the Paperless Office*. MIT Press, Cambridge, MA, 2001.

[162] Phoebe Sengers, Kirsten Boehner, Shay David, and Joseph 'Jofish' Kaye. Reflective design. In *Proceedings of the Conference on Critical Computing (CC)*, pages 49–58, 2005.

[163] N. S. Shami, G. Leshed, and D. Klein. Context of use evaluation of peripheral displays. In *Proceedings of the International conference on Human-Computer Interaction (INTERACT)*, pages 579–587, 2005.

[164] Anoop K. Sinha and James A. Landay. Embarking on multimodal interface design. In *IEEE International Conference on Multimodal Interfaces, Poster*, 2002.

[165] Anoop K. Sinha and James A. Landay. Capturing user tests in a multimodal, multidevice informal prototyping tool. In *Proceedings of the International Conference on Multimodal Interfaces (ICMI)*, pages 117–124, 2003.

[166] Ian Smith. Social-mobile applications. *IEEE Computer*, 38(4):84–85, 2005.

[167] Carolyn Snyder. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. Morgan Kaufmann, San Francisco, CA, 2003.

[168] Lucy A. Suchman. Do categories have politics? the language/action perspective reconsidered. In *Proceedings of the European Conference on Computer-Supported Cooperative Work (ECSCW)*, pages 91–106, 1993.

[169] Pedro A. Szekely. Retrospective and challenges for model-based interface development. In *Proceedings of Design, Specification and Verification of Interactive Systems (DSV-IS)*, pages 1–27, 1996.

[170] Jonathan Trevor, David M. Hilbert, and Bill N. Schilit. Issues in personalizing shared ubiquitous devices. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp)*, pages 56–72, 2002.

[171] Karl T. Ulrich and Steven D. Eppinger. *Product Design and Development*. Irwin McGraw-Hill, Burr Ridge, IL, 2000.

[172] M. van Dantzich, D. Robbins, E. Horvitz, and M. Czerwinski. Scope: Providing awareness of multiple notifications at a glance. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI)*, pages 267–281, 2002.

[173] N. A Van House. Interview viz: Visualization-assisted photo elicitation. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI)*, pages 1463–1468, 2006.

[174] Alexandra Weilenmann. Negotiating use: Making sense of mobile technology. *Personal and Ubiquitous Computing*, 5(2):137–145, 2001.

[175] Mark Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, 1991.

[176] L. Wheeler and H.T. Rois. Self-recording of everyday life events: Origins, types, and uses. *Journal of Personality*, 59(3):339–355, 1991.

[177] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, CA, 2005.

[178] Allison Woodruff and Paul M. Aoki. Push-to-talk social talk. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW)*, pages 409–441, 2004.

[179] Ron B. Yeh, Chunyuan Liao, Scott R. Klemmer, François Guimbretière, Brian Lee, Boyko Kakaradov, Jeannie Stamberger, and Andreas Paepcke. ButterflyNet: A mobile capture and access system for field biology. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 571–580, 2006.

[180] Dongsong Zhang and Boonlit Adipat. Challenges, methodologies, and issues in the usability testing of mobile applications. *International Journal of Human-Computer Interaction*, 18(3):293–308, 2005.

Appendix A

Momento documentation

A.1 Installation

All files can be found at http://m0ment0.com/.

1. Install the desktop client:
   (a) Download and unzip the latest version
   (b) To run in standard mode on Windows, run "momento.exe." If you have not installed Java 1.5, you will be prompted to do so.
   (c) To run in other modes or on other operating systems, use the command-line. For example:

       java -jar momento.jar help

       prints command-line options

       java -mx128m -jar momento.jar -1 nodisco

       runs the application in standard mode.
2. Install the C# mobile client on Pocket PCs
   (a) Download the CAB file and the config file
   (b) Copy the CAB file to the device and open it. Momento should be installed in the "Program Files\momento" directory
   (c) Edit the configuration file and copy it to the "My Documents" directory on the device
   (d) You can also download the whole Visual Studio package.
3. Install, configure, and run the server
   (a) Download and unzip the latest version

(b) Install Apache 1.3.* as a service (windows version)

(c) Install Mysql 4.* (windows version)

(d) Install PHP 4.* (windows version)

   i. Install to c:\php

   ii. Copy \editorial\lib\httpd.conf to \conf

   iii. Copy \editorial\lib\php.ini to C:\windows

   iv. Copy C:\php\sapi\php4apache.dll to c:\php\

   v. Restart apache

(e) Copy files in \editorial\server\web to (Apache directory)\htdocs\editorial

(f) Configure DB parameters in (Apache directory)\htdocs\editorial\editorial.inc (usually you would just need to change db_pswd)

(g) Now you will need to setup Momento and add a Momento admin user, study, and associate the admin and study. Open a terminal and change to the momento dir and run the following:

   java -cp momento.jar editorial.server.Administration createMomento db_root_user db_root_pass c:

   java -cp momento.jar editorial.server.Administration addStudy db_root_user db_root_pass c: study_name study_desc

   java -cp momento.jar editorial.server.Administration addAdminUser db_root_user db_root_pass user_name user_pass

   java -cp momento.jar editorial.server.Administration grantAdminUserStudy db_root_user db_root_pass study_name user_name

(h) Finally, run the server:

   java -cp momento.jar editorial.server.Runner c: db_root_user db_root_pass

A.2 Getting started

To use Momento, you first need to setup a study on a server (see the installation instructions) and download empty study files. Settings will look similar to:
Server: 126.34.43.222
Web root: 80
Study name: my_study
User name: admin
Password: pass

Go to Project→Download Project and enter the settings information. You will be asked to specify a directory: I recommend a directory with the same name as the study ("example_study"). If the directory you specify does not exist it will be

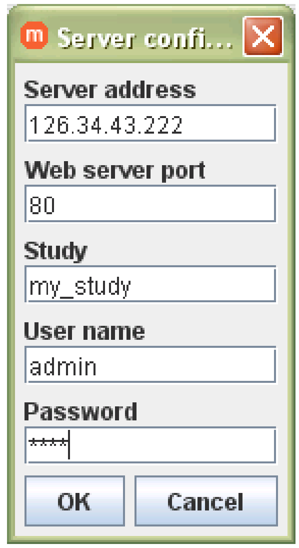


Figure A.1. Configuring the desktop platform to connect to the server. The user name and password should have been created on the server using the Administration program.

created. Then if you have successfully connected to the server the button in the lower right will turn from yellow to green. You can then start adding participants to the project. When participants send events to the server, the events will be passed along and visualized in your client.

If you close Momento, you can re-open the project using Project→Open local project. Settings and data are automatically saved locally, but you can back everything up on the server as well. To do that, use Project→Upload Project.

Keep in mind that all project files are automatically saved in the folder you specify when you open a project, so you do not have to re-save them. When you open a local project, you are opening the version that is stored on your machine. Downloading means copying the version on the server to your machine (which over-writes data stored on your machine). Uploading means copying the version on your machine to the server (which over-writes data stored on the server).

You should only download when you know that someone else has uploaded a more recent version. You probably only need to upload when you know someone else will be looking at the project on a different machine. But just like any computer document, it is best to save your work often, so it is probably good practice to upload once a day.

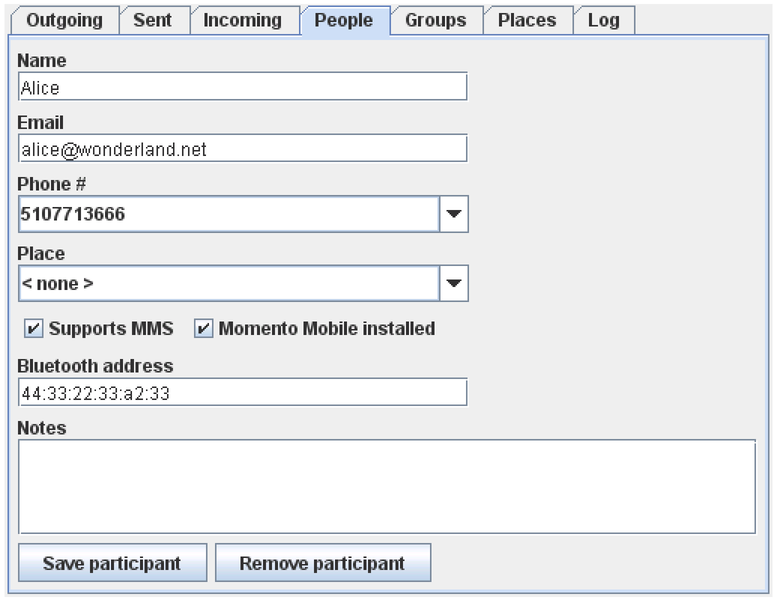


Figure A.2. Configuring a participant in the desktop platform.

A.3 Configuring participants on the desktop platform

You can enter participant information using the participant panel. If you are using the mobile client, it is important that the Bluetooth address field match the Bluetooth address of the participant's device, and that the "Momento Mobile installed" button is clicked. If MMS is disabled, no images will be sent to the participant.

A.4 Configuring rules on the desktop platform

You can configure notifications by:

1. Clicking on the "Outgoing" tab
2. Selecting the participant or group you want to send the message to in the "Apply to participant" tab
3. Typing the text you want to send in the "Txt" box
4. Selecting the file you want sent
5. Click the "Save and send now" button

You will see a blue star appear on the participant's line that indicates that a message was sent to them by Momento.

If you want to schedule recurring message delivery, follow steps 1-3 and

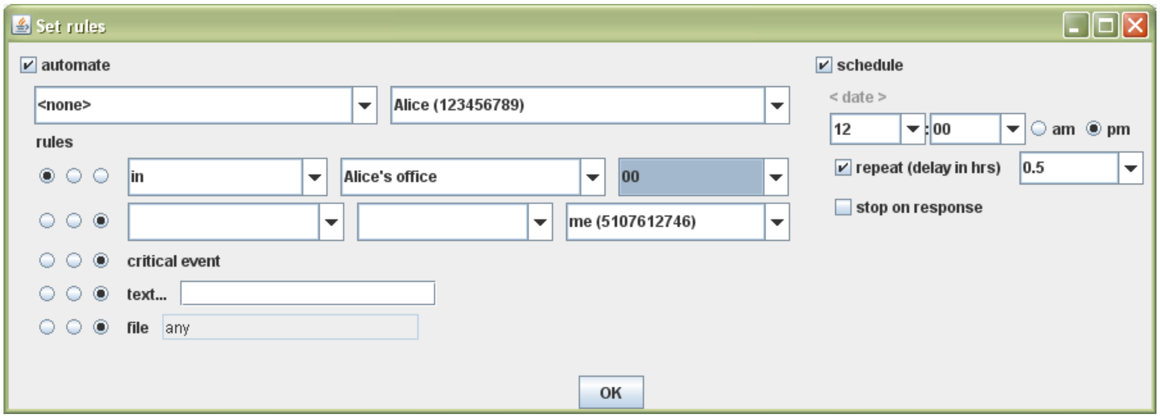


Figure A.3. Example rule. This rule checks to see if participant Alice is in her office at noon and every half-hour thereafter. Note the three columns of radio buttons in the "rules" section have the following meaning: left is sufficient (OR), center is conditionally sufficient (AND), and right is ignore.

1. Click on "Set rules"
2. In the panel that comes up, click the "schedule box"
3. Set the time that you want the message sent in the area below the message box
4. Click on the "repeat delay" button and select the number of hours between repetitions of the message
5. Click on OK
6. Back in the main interface, click on "Save (only)"

A.5 Sending messages to clients from the desktop platform

You can have Momento insert information into the "Txt" box using the following macros: %%PARTICIPANT will be automatically replaced with the participant's name; %%LOCATION the participant's location, and %%PROXIMATE the other participants near this participant.

Note that when sending to a participant running the Momento mobile client, the string in the "Txt" box will do one of two things: 1) if no buttons are defined the text replaces the current description , but 2) if buttons are defined then the client will treat the string as a question. For example, the string "Try harder" would only replace the description string in the mobile client interface, but the string "Are you feeling underwhelmed?,yes,no" instructs the mobile client to create a question with text "Are you feeling underwhelmed?" and with buttons named "yes" and "no". Buttons with these names must be defined in the mobile configuration file.

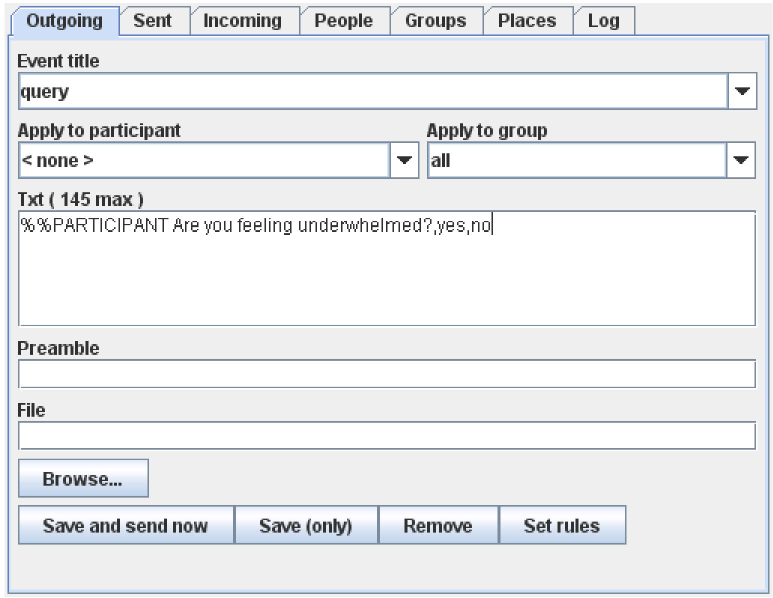


Figure A.4. Configuring an outgoing message. Momento will send this message to every member of group "all". Upon receiving the message, the mobile client will display the question "(participant's name) Are you feeling underwhelmed?" along with buttons with names "yes" and "no" as configured in the mobile configuration file.

A.6 Mobile client

The mobile client can process all incoming messages from Momento: no special configuration is necessary beyond checking the "Momento Mobile installed" box in a participant's configuration settings. Furthermore, messages sent from the mobile client appear on the desktop platform no differently from messages sent from e-mail or SMS/MMS. Please see the installation guide for configuration options for the mobile client.

A.7 Sending messages to the server from standard mobile applications

Standard mobile applications can send text and media attachments to the server via email, SMS/MMS. When sending via SMS/MMS, clients should use the phone number of the server (see section "Connecting the server to the GSM network").

A.8 Desktop platform message sending configuration

You can configure connection information for sending messages via email and via SMS/MMS in the "settings" file for each study. The settings file should include SMTP connection information for sending via email and URLs for gateways capable of sending SMS/MMS data (see section "Connecting the server to the GSM network"). An example "settings" file could include:
EMAIL_SMTP smtp.m0ment0.com
EMAIL_USER gateway@m0ment0.com
EMAIL_FROM gateway@m0ment0.com
EMAIL_PASSWORD password
EMAIL_SSL false
SMS_URL http://128.2.211.152:8800
MMS_URL http://128.2.211.152:8800

A.9 Configuring an email account on the server

The Momento server can receive messages via either email or SMS/MMS. To configure the email account information, include a file called "settings" in the directory containing the server's jar file. An example "settings" file could include:
EMAIL_HOST mail.m0ment0.com
EMAIL_USER gateway@m0ment0.com
EMAIL_PASSWORD password
EMAIL_PROTOCOL pop
EMAIL_SSL false

A.10 Connecting the server to the GSM network

Sending and receiving messages with SMS/MMS requires an interface to an operator service. Momento currently accomplishes this by communicating via a NowSMS gateway with a GSM modem attached to the server. Note that in this scheme, the GSM modem requires a SIM card with a contract on a network. The purpose of the gateway is abstract the process of sending and receiving messages via the cellular modem (which, especially in the case of MMS messages, can be particularly complex).

You can download a trial version of NowSMS[1]. MultiTech makes GSM modems for both EDGE[2] and GPRS[3] networks.

[1] http://www.nowsms.com/
[2] http://www.multitech.com/PRODUCTS/Families/MultiModemEDGE/
[3] http://www.multitech.com/PRODUCTS/Families/MultiModemGPRS/

Note that all messages received by NowSMS should be forwarded to the Momento directory ("editorial_tmp_gsm_files") on the server (please see the NowSMS documentation[4] for instructions). As of October 2006, Momento was able to parse SMS/MMS messages from all major US networks.

Other configurations are possible. First, while NowSMS is currently the most complete gateway on the market, it is expensive. Kannel[5] is an open-source WAP gateway that can be configured to send and receive both SMS and MMS messages. Another option is to use a network services that abstract both the gateway and communication with an operator network, so no GSM modem is required. An example service is jSMS[6]. However, these services are even more expensive that the NowSMS/GSM modem approach.

A.11 Special desktop platform commands

Special commands in the Nokia coding syntax can be used to control the desktop platform in real time. The following commands can be sent to the desktop platform from a mobile phone. Note that these commands send SMS acknowledgements back to the sender:

***#stop# project** Stops the desktop platform from sending any automatic or scheduled events to participants for the given project

***#start# project** Allows the desktop platform to send events to participants for the given project

A.12 Calendar input to the desktop platform

A calendar file can be configured to indicate times to block all events to participants. The calendar file should be stored in the main project directory with the name "participant_schedules" and should have the following structure (this information should appear on one line):

```
##########,location,who [and who ... ],HH:MM,TT,HH:MM,
[M|Tu|W|Th|F|Sa|Su][,r]
```

[4]http://nowsms.com/documentation/
[5]http://www.kannel.org/
[6]http://www.objectxp.com/products/jSMS/

For example, to specify a meeting for participant with number 510-761-2733 at that participant's office with Jim and Chris at 10:00AM every Tuesday and Thursday, write:

5107612733,office,Jim and Chris,10:00,AM,1:00,TuTh,r

Appendix B

Understanding difficulties with mobile iteration: interview guide

The goal of the interview is an open discussion about challenges the interviewee has faced when designing, developing, and evaluating mobile applications. The focus will be on early-stage iterative techniques.

During the interview, we will collect standard demographic information (such as age and gender) and we will address the following questions:

1. What is your current occupation?
2. In what capacity do you currently work on mobile applications (primarily design, primarily development, primarily evaluation)?
3. What percentage of your work focuses on this field?
4. In your career, how many projects have you worked on in this field?
5. On what aspects of these projects did you focus?

Describe your experiences brainstorming and evaluating early-stage (low fidelity) versions of mobile applications.

1. Do you believe this step is important for your work? Why?
2. Describe one aspect of this approach that was successful and one that was not.

Describe your experiences developing mobile applications.

1. What tools have you used? Why?
2. Describe your difficulties/successes with these approaches.

3. Given what you know now, what approach would you take if you were to develop these applications again?

4. Do you have any specific suggestions for future toolkits?

Describe your experiences evaluating these applications in laboratory settings.

1. Do you believe this step is important for your work? Why?

2. What did you try to measure? Which techniques did you use? What was successful? Why? If you were to do it again, how would you change your approach?

Describe your experiences deploying applications in field settings.

1. Do you believe this step is important for your work? Why?

2. What devices did you use? Were the devices your own or owned by the participants? What problems did you have with the devices?

3. What techniques did you use to gather data? Explain any difficulties with these techniques. Also explain some successes of these techniques. If you were going to run another study, what techniques would you use? How?

4. Overall, was the deployment a success? Why? Explain any issues with the application that were not due to the device.

Appendix C

Gathering implicit data with J2ME: lessons learned from the AwarenessBoard study

Our experience with this project indicated that it may not be possible to gather real-time, implicit data via a J2ME application (MIDlet) installed on most phones. To gather real-time, implicit data, the application must be able to use a network connection intermittently to send data back to the server without interrupting the user. However, security settings on most mobile phones make it such that for third party applications (such as those developed by research developers), users are forced to respond to a prompt whenever a new network session is requested.

J2Me applications are assigned one of three security levels that governs the extent to which different libraries can be used autonomously. Unsigned applications are always given the least-secure, most-restricted *default* profile. In this profile, many libraries are completely disabled while others usually require user permission before use. An application signed with a certificate installed on the phone is given a *third party* profile that usually allows the application to use many more libraries than the *default* profile, but still requires user input before the application is able to use certain libraries (usually those that would cost the user money, such as network and media messaging libraries, as well as those that facilitate networking with nearby devices, such as Bluetooth libraries). Developers can gain this level of security by creating their own certificate, installing it on a device, and signing their application with that certificate before deploying it to the device. Finally, cellular *operators* can sign applications with a hidden certificate that removes all restrictions on the signed application. This security level is only available to developers who enter into an explicit agreement with a cellular operator. Note that other means of achieving *operator*-level application permissions are possible, but only by reprogramming and redeploying the mobile device's kernel or by purchasing the device directly from a manufacturer not affiliated with an operator. Because manufacturers and operators collude in most countries, this effectively means buying from third-world manufacturers (some researchers have reported buying such systems from street vendors in Chinatown, San Francisco, California).

Given the difficulties of obtaining *operator*-level security profiles, most developers use *third party* settings. However, there are many reasons why security levels might vary for one or more libraries in third party signed applications. For example, applications deployed locally rather than via OTA (over-the-air transmission, or downloaded via a cellular network onto the device) tend to have overall lower permissions settings. Security levels can also vary by operator, device, and even operating system version. This means that it is practically impossible to anticipate exactly how an application will operate on a user's device unless the developer installs the application him/herself.

In our deployment, we signed our applications with *third party* settings. Again, this required generating and installing our own certificate and then signing our application with that certificate. We found that we were able to get the level of permissions necessary to support automatic, implicit network use only if the application was installed OTA. While this meant that deployment was more expensive than we anticipated, we were still hopeful that the application would operate properly. Unfortunately, we found that the operating system itself required that users select an access point to use in any network session *even if there were only one access point configured on the device*. After looking into this issue more thoroughly, we found that this is the case on most Symbian operating system versions installed on most devices. We were not able to get around this issue for our deployment, meaning that users had to explicitly answer this question on the device for the application to send data to the server. This had a large negative impact on data collection.

This seemingly innocuous operating system dialog more-or-less destroys any possibility of implicit data capture on J2ME applications. Because of this, we abandoned J2ME as a development platform in favor of Windows Mobile. While Windows Mobile is not as widely used as J2ME, it has not been crippled by security concerns.

Appendix D

Tangible objects collected in the festival diary study.

Figure D.1. Tangible objects collected in the festival diary study.