HUMAN–COMPUTER INTERACTION, 2008, Volume 23, pp. 47–99
Copyright © 2008 Taylor & Francis Group, LLC
ISSN: 0737-0024 print / 1532-7051 online
DOI: 10.1080/07370020701851086

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Exiting the Cleanroom: On Ecological Validity and Ubiquitous Computing

Scott Carter[1], **Jennifer Mankoff**[2], **Scott R. Klemmer**[3], and **Tara Matthews**[4]
[1]*FX Palo Alto Laboratory, Inc.*
[2]*Carnegie Mellon University*
[3]*Stanford University*
[4]*IBM Almaden Research Center*

ABSTRACT

Over the past decade and a half, corporations and academies have invested considerable time and money in the realization of ubiquitous computing. Yet design approaches that yield ecologically valid understandings of ubiquitous computing systems, which can help designers make design decisions based on how

Scott Carter is a research scientist at FX Palo Alto Laboratory; his work focuses on tool support for early-stage experimentation with ubiquitous computing applications. **Jennifer Mankoff** is an Assistant Professor in the Human–Computer Interaction Institute at Carnegie Mellon University; her research interests are in tools and techniques for developing iterative prototypes of ubiquitous computing applications, accessible technology, and Internet-scale prototyping and evaluation. **Scott Klemmer** is an Assistant Professor of computer science at Stanford University; his research passion is enabling designers and other innovators to create interactive media that integrates the physical and digital worlds. **Tara Matthews** is a computer scientist at the IBM Almaden Research Center, with interests in peripheral displays, glanceability, evaluation, multitasking, accessibility, and ubiquitous computing.

CONTENTS

1. INTRODUCTION
2. FIELDWORK WITH UBICOMP DEVELOPERS
2.1. Peripheral Displays
Method
Results
Discussion
2.2. Mobile Applications
Method
Results
Discussion
2.3. Integrating Physical and Digital Interactions
Method
Results
Discussion
2.4. Challenges for Ubicomp Evaluation
3. STRATEGIES FOR ECOLOGICALLY VALID DESIGN
3.1. Observation
Experience Sampling
Diary Studies
3.2. Lightweight Prototyping and Iteration
3.3. Functional Prototypes
3.4. Controlled Evaluations
3.5. Field Experiments
3.6. Summary
4. IMPLICATIONS
4.1. Conversations With Materials
4.2. Prototyping for Evaluation
4.3. Supporting In-the-World Evaluation
4.4. Support for Machine Learning and Sensor-Based Interaction
4.5. Data Sparsity
5. CONCLUSIONS

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

systems perform in the context of actual experience, remain rare. The central question underlying this article is, What barriers stand in the way of real-world, ecologically valid design for ubicomp? Using a literature survey and interviews with 28 developers, we illustrate how issues of *sensing* and *scale* cause ubicomp systems to resist iteration, prototype creation, and ecologically valid evaluation. In particular, we found that developers have difficulty creating prototypes that are both robust enough for realistic use and able to handle ambiguity and error and that they struggle to gather useful data from evaluations because critical events occur infrequently, because the level of use necessary to evaluate the system is difficult to maintain, or because the evaluation itself interferes with use of the system.

We outline pitfalls for developers to avoid as well as practical solutions, and we draw on our results to outline research challenges for the future. Crucially, we do not argue for particular processes, sets of metrics, or intended outcomes, but rather we focus on prototyping tools and evaluation methods that support realistic use in realistic settings that can be selected according to the needs and goals of a particular developer or researcher.

1. INTRODUCTION

In the 15 years since Weiser (1991) introduced ubiquitous computing (ubicomp) as a goal, the field has made great strides in terms of system building, but with a few notable exceptions such as Abowd et al. (2000b) and Moran, Chiu, and van Melle (1997), there has been a dearth of iteration and evaluation. For example, Kjeldskov and Graham's (2003) review of mobile human–computer interaction (HCI) systems found field evaluations in only 19 of 102 pieces of published work, and 4 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.

In this article, we suggest challenges and opportunities for ecologically valid design of ubicomp, based on two types of data. First, we draw on fieldwork by the authors with 28 developers in three subfields of ubicomp that, together, flesh out the space of ubicomp applications. Second, we draw on a literature survey of ubicomp systems with the intent of broadly understanding the general state and particular successes of iterative, ecologically valid design and evaluation in ubicomp. These two bodies of evidence are synthesized in a view of ubicomp systems along three dimensions: system properties, challenges to ecologically valid design, and stages of iterative design.

This article offers two contributions. First, our fieldwork and literature synthesis lead us to articulate what we see to be five central challenges for ecologically valid design in ubicomp. Second, we suggest research strategies and highlight improved methods and tools to address these challenges.

We characterize ubiquitous computing as an approach to designing user experiences that, to use Anderson's (1994) phrase, is integrated into the "practical logic of the routine world" (p. 178). 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. They can meet these goals by integrating seamlessly with other successful artifacts. In this way, ubicomp applications can, as Weiser (1991) wrote, "weave themselves into the fabric of everyday life until they are indistinguishable from it" (p. 94). For example, although many have lauded the idea that computers will replace paper,

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

in *The Myth of the Paperless Office,* Sellen and Harper (2001) showed that users' work practices are much more successful, and much more subtle, than a naïve techno-utopian perspective might suggest. Figure 1, from Mackay's (1998) 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. In summary, ubicomp applications that *augment* a user's existing practices can often be more successful than those that seek to supplant them (Dourish, 2001; Klemmer, Hartmann, & Takayama, 2006).

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 article:

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

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Figure 1. **Air traffic controllers work with paper flight strips, from Mackay, Fayard, Probert, and Médini's (1998) research. Prior work to replace the physical world of air traffic controllers with a graphical user interface had "been rejected by the controllers." Mackay et al. found that "automation need not require getting rid of paper strips. We suggest keeping the existing paper flight strips as physical objects, with all their subtlety and flexibility, and augmenting them directly by capturing and displaying information to the controllers" (p. 98). Copyright 1998 by Wendy E. Mackay. Reprinted with permission. Color versions of all figures are available at http://www.cs.conv.edu/~jmankoff/tochi-ubicom-eval.html.**

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 multitasking 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 (Gonzalez & Mark, 2004). Ubicomp applications can benefit from being sensitive to these tasks or supporting this multitasking process. Applications in the subarea of ubicomp called peripheral displays are often used in multitasking 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., Churchill, Nelson, Denoue, Helfman, & Murphy, 2004) and systems supporting coordination among small, collocated working groups (e.g., Carter, Mankoff, & Goddi, 2004).
- ***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 personal digital assistants, 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 comprises "real-world" use of a system, is challenging to achieve because ubicomp applications tend to support not only many aspects of a single activity but also the interaction of multiple activities. The focus of this article is addressing the challenge of achieving ecological validity. In particular, this article focuses on evaluation

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

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., Nielsen, 1989) and laboratory studies (e.g., Rubin, 1994). 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, including adoption, sparsity, and critical mass (e.g., see Grønbæk, Kyng, & Mogensen, 1992; Grudin, 1994; Herbsleb, Atkins, Boyer, Handel, & Finholt, 2002; Olson & Olson, 2000). Without addressing ecological validity, developers risk making and evaluating "a representation without sufficient knowledge of how it actually would work," what Holmquist (2005) called "cargo cult design" (p. 50).

As an example of the value of ecological validity, consider the design process of CareNet, an ambient display connecting elders with their families (Consolvo et al., 2004). CareNet was deployed in a field experiment that employed activity sensing using Wizard of Oz. Wizard of Oz is an early-stage evaluation technique in which a person (the "wizard") simulates a task that, once implemented, would be handled by a computer (Dahlbäck, Jönsson, & Ahrenberg, 1993; J. F. Kelley, 1984; Maulsby, Greenberg, & Mander, 1993). The researchers found that, to succeed, a system would need to utilize "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" (Consolvo et al., 2004, p. 13). 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 researchers 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" (p. 11). The value of ecologically valid evaluations is evident in other research systems as well. For example, in a yearlong field trial of a Digital Family Portrait, another health display, Rowan and Mynatt (2005) found that "behavior shifted gradually with the changes in the seasons" (p. 529). 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.

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

The rest of this article is organized as follows. For purposes of exposition, we organize our discussion of ecologically valid design into four areas: observations, prototyping (including both early-stage and functional prototypes), controlled evaluations, and field experiments. Although we divide our text, we try to acknowledge the fluid way in which a researcher may select from, move between, or combine these areas. Section 2 describes fieldwork by the authors with 28 developers in three key subfields of ubicomp. Section 3 synthesizes ubicomp literature and our fieldwork, presents challenges to ecologically valid design, and then describes how they affect different aspects of the four areas. These two sections provide the background for Section 4, which articulates implications for future research in needfinding, prototyping, and evaluation. Most notably, we show that ecologically valid design is challenging because of the centrality of sensing and scale to the ubicomp experience, and we argue that an important direction for methodology is the creation of techniques for gathering longitudinal data without requiring an exponential increase in labor. Although we recognize that research and development are distinct enterprises, the insight underpinning this article is that to achieve ecological validity, tools and methods need to move further into the practical world.

2. FIELDWORK WITH UBICOMP DEVELOPERS

To better understand the challenges to ecologically valid design in ubicomp, we present the results of interviews with 28 developers in three subfields of ubiquitous computing: peripheral displays, mobile systems, and augmented paper user interfaces. 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 includes technologies that support both individual and group tasks. Through fieldwork 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 research.

In presenting the findings of our field work, 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. To protect interviewees' privacy, we illustrate the issues found in the interviews using topically similar systems developed by noninterviewees for illustration purposes.

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. Peripheral Displays

Peripheral displays are tools that enable glanceable and noninterruptive access to information. Many are intended to be understood with minimal training, though some displays become peripheral only after extensive use. 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, whereas their noninterruptive nature minimizes the extent to which they distract from that activity. An example of a peripheral display is Pinwheels (see Figure 2), which maps the spin of pinwheels to the rate of change of a variety of information sources (Ishii, Ren, & Frei, 2001).

Figure 2. **The Media Lab's Pinwheels are a peripheral display that uses rotational velocity of actuated pinwheels to represent stock market trends (Ishii, Ren, & Frei, 2001; Wisneski et al., 1998). Copyright 2001 by MIT Media Lab Hiroshi Ishii. Adapted with permission.**

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Method

Matthews (2007a) conducted interviews with 10 peripheral display designers. Six of the participants were academics, and 4 were industrial researchers. Three of the participants were primarily designers, 3 were primarily developers, and 4 were both. Three participants had built one or more toolkits relevant to peripheral displays.

Interviews were conducted in person when possible and over the phone otherwise. Each interview began with an explanation of our goal: to determine ways to support the process of designing, implementing, and/or evaluating peripheral displays. We then asked predetermined questions that helped us explore the difficulties creators faced at each stage.

Results

Participants discussed challenges to both prototyping and evaluating peripheral displays. They reported developing costly functional prototypes early on because they doubted that prototypes that simply "looked" like their intended displays could elicit useful user feedback. Evaluation was sometimes difficult because attention and information awareness are highly sensitive to small changes such as those that may be caused by lightweight prototypes and observation.

Early on in the iterative design process, Matthews' (2007a) interviewees found it difficult to determine how their study participants used peripheral information. Peripheral information often only subtly influences work practice, making use difficult to observe. At the same time, asking study participants to self-report on their use of peripheral information resulted in feedback that some interviewees did not trust. This was because both self-reports and observations may have changed the way an end user interacted with peripheral information by bringing it to his or her focal attention. Possibly because of difficulties encountered when needfinding, one interviewee had trouble "justifying [the] existence" of peripheral displays.

When a need was identified, and prototyping began, the first issue interviewees encountered was deciding among the many design options. One participant said,

> I think it's frustrating because there are so many options for designing the information. Literally in some instances, there are millions of options and you're never going to be able to systematically test all of those. … If you could find ways of assessing large amounts of options quickly, that would be fantastic.

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

As a consequence, increased early-stage iteration is needed. Yet interviewees described difficulties rapidly achieving "as real of an experience as would suffice for your data collection needs." For example, one interviewee felt that prototypes that simply *looked like* his planned display would not suffice, eliminating a whole class of low-cost prototyping techniques. He was developing a scent-based display that led to unique usability issues that would have been difficult to discover without experiencing the smells (such as the fact that a smell "stays around for a while …"). Participants also commented that it was difficult to create nondistracting, glanceable prototypes using lightweight prototyping methods. One reason for this was a lack of design knowledge. Matthews's ongoing work on glanceability (see Matthews, Czerwinski, Robertson, & Tan, 2006b) may help to address this.

Participants felt that "the real value in many of these systems is only apparent longitudinally." Thus, participants were interested in building and deploying functional prototypes as rapidly as possible. Our interviewees expressed a need for tools that support building applications that use multiple output modalities (physical, graphical, or audio), that use input from sensors and that depend on distributed input and output. These issues also play out in the literature. In one publication, the authors report spending about one person/year developing a working display (Heiner, Hudson, & Tanaka, 1999).

The literature indicates that the most common evaluations of peripheral displays have been controlled lab studies, usually of the dual-task variety. However, despite the relative popularity of this approach, many participants told us that designing a realistic lab study was difficult for them. One interviewee said that "evaluation is the hardest part. How do you evaluate a peripheral display—you can't do a typical lab/usability study ...." Participants reported difficulties not only with how to structure a lab study but with what to look for when running it. One participant said,

> I would have liked [to use some metrics], if I had known what metric to look for. That I guess is where I felt there was a lag in the project. … At least we did not know of enough, or of any psychological theory that could come and assist us here. Something that you could measure and then predict about longitudinal effects.

Another participant pointed out that knowing what to measure, and how to measure it, were separate challenges: "We could really use methodology to evaluate peripheral displays in terms of usefulness, desirability, and distraction. … We never had it and [so] it was hard." Even so, controlled experiments were viewed as important because, as the same interviewee pointed out, "we had to implement a working prototype and deploy it in people's work place. If we had found that it was all wrong, we would have had to throw away all that work."

In the end, participants tended to feel that the best way to learn whether and how a particular peripheral display was successful was a situated, long-term deployment. One participant expressed this need:

> [Evaluation] is so hard when you are talking about peripheral awareness because how are you going to prove that it is successful, except if people after two years are still using it? … But you cannot test it after a month because it is like a language: you have to learn and you are not learning it in an intellectual way, you are learning it in a sympathetic way. You are getting it internalized gradually.

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Interviewees who had conducted field studies lasting several weeks reported that unobtrusive observation and system maintenance were the two most crucial problems when deploying and studying peripheral displays. For example, one participant mentioned difficulties deploying a display in an unobtrusive way to a participant's home:

> "A cord … is not going to be accessible … in the home. So [the display] needs to have wireless communication built into it and 802.11 is a far too heavy weight. We were looking into using a little AM radio transmitting and receiving pair that went to a box that plugged by USB into your computer. But, then you still would have to have your computer on all the time, so it is not a perfect solution.

Interviewees also found it difficult to keep deployments running because of the extensive maintenance required:

> [We deployed our systems for a] couple of weeks to a couple of months. [We stopped using them] usually because they stopped working. There was no planned undeployment. … It was sort of, they would stop working and you would reboot them and they would start again, or you would have to clean up something and so you would accidentally unplug the thing they were plugged into.

Another interviewee lamented that while it was possible to update software in the field, "you can't download hardware," making it difficult to recover from device breakdowns.

Finally, interviewees found it difficult to gather quantitative data in situ while remaining unobtrusive, which is particularly critical with peripheral displays, for which nondisruptiveness is a design goal. This concern led interviewees to rely instead on post hoc survey data. However, these data were problematic: One interviewee who gathered e-mail survey data found responses "not satisfying" and did not "fully trust" the answers because the interviewee believed study participants would have difficulty recalling the dis-

play's effects on their behavior. These problems with post hoc self-reporting are well-known issues in the experimental psychology literature.

Discussion

The central problem facing developers of peripheral displays is that metrics for success are not well defined. One participant summarized this issue saying that although "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 to 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, Fong, Ho-Ching, and Mankoff (2006c) conducted needfinding interviews and sketch studies that led to the IC2Hear sound awareness display. In this study, the sketches gave users semiconcrete display ideas to discuss. The rough nature of the sketches encouraged critiques and suggestions, improving the prototypes created based on interview results. Similarly, Sengers, Boehner, David, and Kaye (2005) instructed participants to "reflect on aspects of their current relationship and technology use within that relationship, and … sketch novel designs for communication devices for couples to use" (p. 54).

Researchers are deriving metrics and design guidelines for peripheral displays. Work by Mankoff et al. (2003) adapts 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 that go beyond efficiency and ease of use. McCrickard, Chewar, Somervell, & Ndiwalana (2003) have investigated ways to identify relevant metrics and evaluation strategies for peripheral displays. In particular, they utilized a design model for classifying different types of peripheral awareness systems along the dimensions of interruption, reaction, and comprehension. Finally, Matthews, Rattenbury, and Carter (2007b) derived criteria, including appeal, learnability, awareness, effects of breakdowns, and distraction, as well as guidelines for evaluations from past literature and a user-centered activity theory framework.

The Context of Use Evaluation of Peripheral Displays (CUEPD) method, developed by Shami, Leshed, and Klein (2005), captures the context of use

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

through user scenario building, enactment, and reflection. Researchers have found that designers can use CUEPD to improve future designs, once they have a working prototype. This method increases 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 (Greenberg & Fitchett, 2001), iStuff (Ballagas, Ringel, Stone, & Borchers, 2003), and d.tools (Hartmann et al., 2006). Furthermore, Matthews, Dey, Mankoff, Carter, and Rattenbury's (2004) 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 (van Dantzich, Robbins, Horvitz, & Czerwinski, 2002). 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, Mankoff, and Landay (2003) 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, Czerwinski, Robertson, and Tan (2006b) 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, Venolia, Jancke, and Gupta (2002) was particularly successful. Sideshow is a graphical peripheral display of various information streams (e.g., meetings, e-mail, instant messaging, coworker presence, traffic, weather). During a nine month period, 22 new versions of Sideshow were released with bug fixes and new features. The updates were made based on a constant dialog with users, who submitted bug reports and e-mail feedback. For example, laptop users requested an "offline" mode that showed stale data. Although hesitant to show outdated information, designers added this feature and got positive feedback from users. This successful iteration process was facilitated in large part by a focus on making Sideshow easy to maintain and update. Sideshow had an ad-

vantage 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.

2.2. Mobile Applications

Mobile applications are those deployed to personal devices that people carry from place to place (see Figure 3 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

Carter (2007) conducted interviews with nine designers of mobile applications. We focused on developers who had deployed applications to personal digital assistants and mobile phones. Six participants held research positions; the other three worked in nonresearch, industry positions. Three of the partic-

Figure 3. **Functional prototype of the Scribe4Me system, which provides an on-demand transcription service for the deaf. By pressing "What happened" (a), the user causes the previous 30 seconds of audio and an image to be sent to a remote wizard (b), who sends back a transcription (c) (Matthews, Carter, Fong, Pai, & Mankoff, 2006). This figure is adapted from "Scribe4Me: Evaluating a Mobile Sound Transcription Tool for the Deaf," by T. Matthews, S. Carter, C. Pai, J. Fong, and J. Mankoff, 2006, *Proceedings of the Ubicomp 2006 International Conference on Ubiquitous Computing,* 163. Copyright 2006 Springer. Adapted with permission.**

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

ipants were primarily designers, three were primarily developers, and three were both. Participants had designed between two and four mobile systems over the last 1 to 3 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 nonmobile designs as well, and ones that should be overcome to promote needfinding.

Our interviews verified what Kjeldskov and Graham (2003) 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 pointed out that many developers conducted 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 designers working closely with developers that variously included extended participant observations, interviews and analysis of collected data, and diary studies. For example, Horst and Miller (2005) 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 and Aoki (2004) lived with teenagers for one week to understand how teenagers use push-to-talk technologies. This experience provided 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

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

difficult to contrive realism using lightweight prototypes. Instead one developer concentrated on mock-ups of his display that he then used in a cognitive walkthrough (similar methods, such as heuristic walkthroughs, have also been used in the literature; e.g., Kjeldskov et al., 2005). Using this approach, a developer could "find the really big and the really small" problems with the design without worrying about "trying to get the user to imagine" that he or she was 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 (2003) showed 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 person commented, "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 previously described, 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 anymore or you will have a different payment plan" and "sometimes it is hard to find out what [the operator's] limits [are] for various features … like data limits on messages." As an example, in the Scribe4Me system (Matthews, Carter, Fong, Pai, & Mankoff, 2006a), which sends audio and pho-

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

tographs across the MMS network to provide transcriptions for the deaf (see Figure 3), on rare occasions we encountered delays of up to nine hours when messages had to cross between service providers.

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 were overcome, developers encountered evaluation challenges similar to those in needfinding studies. For example, in their study of a mobile presence awareness device for ski instructors, Weilenmann (2001) found that "the observer's task is difficult—it is simply not possible to be everywhere at the same time." As a result, they used participant observations and focus groups to evaluate the awareness device. The developers we interviewed had similar concerns and chose either to run 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 of 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 because of 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.

Carter's (2007a) participants verbalized a concern about the difficulty of collecting ecologically valid data with lightweight mobile prototypes. Others have reported similar concerns. For example, Rudström, Cöster, Höök, and Svensson (2003), in a paper prototype study of a mobile social application, found that participants had difficulty reflecting on how their use of the application would change if they actually were mobile and using an interactive system. Carter, Mankoff, and Goddi. (2004) 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 proved difficult for them.

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, Tamminen, Roto, and Kuorelahti (2005) articulated an important shortcoming of lab studies in the mobile domain: 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 to devise more rapidly buildable approximations of a system that can be used to move controlled studies into the field. Kjeldskov, Skov, Als, and Høegh (2004b) re-created 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 (2004a) 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 (2006) controlled field experiment with 14 biologists of the ButterflyNet system, a device ensemble comprising a mobile device and an augmented paper notebook, the key insight was to use a handheld machine running Microsoft Windows XP® software to simulate the features of a future digital camera (see Figure 4).

Because of the large 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 (2006) 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 among teenagers, Grinter and Eldridge (2001) talked about using diary studies because direct observation "would be impractical" and "teenagers were hesitant about being directly observed" (p. 442). Palen, Salzman, and Youngs (2000) used a voice-based diary to study mobile phone calls, finding design issues with public mobile phone use. The PlaceLab group at Intel Research Seattle ran an experience sampling study to understand how factors such as activity and mood affect location disclosure in mobile applications and used this data in the design of a social location disclosure service applica-

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Figure 4. **OQO can be a more easily programmable proxy for a future smart digital camera. With the smart camera, users can perform on-the-spot annotations of photos by marking on the LCD screen with a stylus (Yeh et al., 2006). The smart camera also communicates wirelessly with the pen, offering real-time visual and audio feedback for in-the-field interactions. This smart camera was prototyped with an OQO handheld running Microsoft Windows XP software with a Webcam affixed to the back. This figure is reprinted from "ButterflyNet: A Mobile Capture and Access System for Field Biology," by R. B. Yeh, C. Liao, S. R. Klemmer, F. Guimbretière, B. Lee, B. Kakaradov, J. Stamberger, and A. Paepcke, 2006, *Proceedings of the CHI 2006 Conference on Human Factors in Computing Systems*, 574. Copyright 2006 ACM, Inc. Reprinted with permission.**

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

tion (Consolvo et al., 2005; Smith, 2005). Abowd et al. (2005) 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 Microsoft Internet Explorer® and for Firefox) 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 (http://www.forum.nokia.com/python/) or Mobile Processing (http://mobile.processing.org/), 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 (Benford et al., 2006; Fleck et al., 2002).

2.3. Integrating Physical and Digital Interactions

A primary goal of ubiquitous computing is the creation of systems that augment the physical world by integrating digital information with everyday physi-

cal 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 (1998a) 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 (see Figure 1).

Method

Klemmer (2004a) conducted structured interviews with nine researchers who have implemented tangible user interfaces. Four of the interviewees worked in academia; the other five worked in industrial research. Four researchers had experience developing high-fidelity tangible user interfaces prior to the project discussed in the interview. For these groups, the project we discussed was a continuation of work in this area. This next step was exploring an alternate point in a design space, exploring richer interactions, delivering greater use value, or exploring lower complexity.

Questions addressed general system design, planning and organizational structure, software design, user and system evaluation, and difficulties in design and implementation (Klemmer, 2004a, Appendix C). These interviews were conducted in person at the workplaces of researchers (three), over the phone (one), or via e-mail (five).

Results

The primary challenge developers faced was that acquiring and abstracting physical input—dealing with sensing—required a high level of technical expertise and a significant time commitment. Interviewees explained that sensing-based input technologies such as computer vision do not always behave as planned. Consequently, they felt it was important to design systems robust to occasional errors and input ambiguity and to provide feedback so that users could diagnose and help recover from system errors. In one interviewee's words, "the sensing hardware is not perfect, so sometimes we had to change interactions a bit to make them work in the face of tracking errors."

At the needfinding stages, Klemmer (2004a) found a diversity of approaches. Some interviewees were "exploring" or simply building a "passion-driven device," whereas others based their work on ethnographic or diary studies. Others simply spoke with a single user, who may or may not have inspired the technology being developed.

Prototyping was an important medium for exploration among the interviewees. They reported using prototypes to understand interaction scenarios

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

and to gain fluency with the media they were using for development. Two of the interviewees began with paper prototypes, often trying out different scenarios to understand the interactions required before writing code. "The paper prototypes helped us understand the space considerations/constraints … helped us work through the scenarios." One of these researchers also used physical objects (without computation) "to get an idea of what it would feel like to use our system." The remaining seven interviewees began prototyping with technologies and tools that they were familiar with or that had a low threshold and only later explored less familiar or higher threshold tools.

Issues raised by the interviewees pointed to a need for better tools. Interviewees reported that they were forced to implement extensive system redesigns when making straightforward interface changes such as switching between input technologies (e.g., a camera and barcode reader). A result of this problem was that "the code was way too complex at the end of the day" because there were "a lot of stupid problems" such as temporary files and global variables that inhibited reliability and malleability. In addition, this fieldwork found that each development team was creating an architecture, a set of library components including custom software for acquiring and abstracting input from each new piece of hardware, and an application (though the developers did not generally describe their work with such an explicit taxonomy). The basic event-based software design patterns were uncannily similar across many of these systems, and yet, at the time they were built, no tool existed that could save developers that effort.

A few interviewees chose to evaluate their interaction design through comparative studies (either to a "somewhat comparable GUI [graphical user interface]" or to "several alternatives"). Others chose not to run any studies ("It would have been a pile of work"). Still others ran many informal "grab your colleagues" tests or demos. In addition to understanding the end-user experience, interviewees wanted to develop a better understanding of use from a system perspective. They wanted to be able to find out answers to questions such as "Which sensors did they use? [Did they use them] the way you think or something else completely?"

A few interviewees also reported conducting fieldwork using their systems. One major motivation for this was to increase robustness and to find problems, including software bugs, recognition ambiguities and errors, and usability errors. One interviewee told us that he "put it up, and ran it for about six months in two or three locations in the building." To evaluate the robustness of the system, he then watched for failure modes. "These failure modes helped drive further development. This failure mode analysis is key." Another told us, "We were worried about robustness. So I made a prototype and left it in the hall for months."

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Discussion

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.

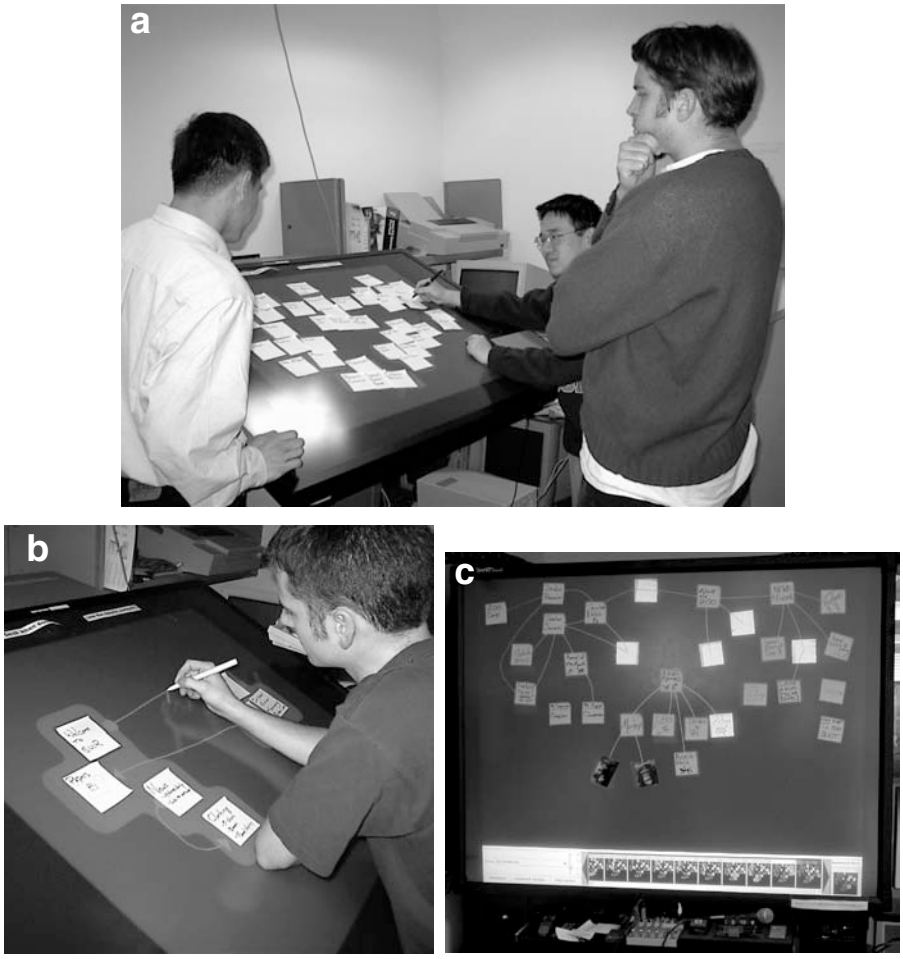
In addition to Mackay, Fayard, Frobert, and Médini's (1998a) fieldwork with air traffic controllers, other researchers have conducted needfinding studies of tangible interfaces that successfully translated to prototypes. In their study of Web designers, Newman, Lin, Hong, and Landay (2003) found that 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 (see Figure 5). Also, Yeh et al.'s (2006) fieldwork led to the creation of tools to support data capture for biologists working in the field.

Prototyping was beneficial to interviewees. Our results demonstrate that the interviewees' prototypes helped them to learn and that the interviewees' different approaches provided different insights. We 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 may be useful for ubicomp (Hill, 1986). Furthermore, a significant difficulty in program debugging is the limited visibility of application behavior (Détienne, 2001). 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 et al. (2001) evaluated Outpost with professional Web designers. Participants were asked to "speak aloud" about their experiences while they completed an information architecture design task. Fitzmaurice, Ishii, and Buxton (1995) implemented and evaluated a tangible interface to Alias Studio, a high-end three-dimensional modeling and animation program. The evaluation found that users rapidly learned how to perform complex operations. Finally, McGee, Cohen, Wesson, and Horman (2002) 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 sticky 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.

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Figure 5. **The Designers' Outpost integrates wall-scale, paper-based design practices with novel electronic tools to better support collaboration for early-phase design (Klemmer, Newman, Farrell, Bilezikjian, & Landay, 2001). With Outpost, users collaboratively author Web site information architectures on an electronic whiteboard using physical media (sticky notes and images), structuring and annotating that information with electronic pens. This interaction is enabled by a touch-sensitive electronic whiteboard augmented with a computer vision system. Early pixel and physical form mock-ups of Outpost (a,b) helped the researchers flesh out the interaction techniques used in the final version (c). Figure (c) is reprinted from "Where Do Web Sites Come From?: Capturing and Interacting With Design History," by S. R. Klemmer, M. Thomsen, E. Phelps-Goodman, R. Lee, and J. A. Landay, 2002, *Proceedings of the CHI 2002 Conference on Human Factors in Computing Systems,* 2. Copyright 2002 ACM, Inc. Reprinted with permission.**



Extended field deployments of tangible interfaces are rare, but some evidence shows that they can yield important insights. Maldonado, Lee, Klemmer, and Pea (2007) 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 58 design students and recorded more than 4,000 pages of authored content. Their results showed that their tool enabled new behaviors, including reflection on design process. Improved prototyping tools and evaluation methods have the ability to lower the threshold for such valuable deployments.

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

2.4. Challenges for Ubicomp Evaluation

Our interviews revealed that designers of ubicomp applications struggle with ecological validity throughout the design process. For example, Figure 6(c) shows a system, Hebb (Carter, Mankoff, & Goddi, 2004), 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 (Carter & Mankoff , 2005a), 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. (2002) discuss some of the issues that arise from inferencing, including recovering from mistakes (illustrated in Figure 6a), clearly articulating the target of a command, and clearly indicating to whom the system is attending. These represent core usability issues. Possible solutions, such as mediation techniques like reaction and chore (Mankoff, Hudson, & Abowd, 2000), can be tested only if recognition errors and ambiguity occur at realistic rates during evaluation. In addition, low accuracy sensing and inferencing can have a huge negative 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

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Figure 6. (a) An 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. This figure is reprinted from "Momento: Support for Situated Ubicomp Experimentation," by S. Carter, J. Mankoff, and J. Heer, 2007, *Proceedings of the CHI 2007 Conference on Human Factors in Computing Systems*, p. 131. Copyright 2007 ACM. Reprinted with permission. (b) An in/out board asks if it has correctly sensed that Gregory Abowd is leaving. Interactive confirmation is one technique of dealing with potential errors. This figure is reprinted from "Distributed Mediation of Ambiguous Context in Aware Environments," by A.K. Dey, J. Mankoff, G. Abowd, and S. Carter, 2002, *Proceedings of the UIST 2002 Conference on Human Factors in Computing Systems*, p. 128. Copyright 2002 ACM. Reprinted with permission. (c) Hebb, a 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 (Carter *et al.* 2004). (d) The Peripheral Display 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. This figure is adapted from "A Toolkit for Managing User Attention in Peripheral Displays," by T. Matthews, A.K. Dey, J. Mankoff, S. Carter, and T. Rattenbury, 2004, *Proceedings of the CHI 2004 Conference on Human Factors in Computing Systems*, p. 247. Copyright 2004 ACM, Inc. Adapted with permission.

in the myriad settings where tasks may occur and because data collection for sensing purposes may be difficult. For example, in any system that depends on a large corpus of labels, using sensed data for inferencing will be especially difficult to prototype if data are sparse. Overcoming this challenge often requires running evaluations over large amounts of time, people, or places.

Critical mass. For ubicomp applications built to involve many tasks, places, people, or devices, reaching critical mass along the relevant dimension is important to ecological validity. This requires prototypes to scale robustly. It affects evaluation because difficulties such as adoption by many people, such as with Hebb (Carter, Manko, & Goddi, 2004), or unanticipated interference with existing activities may arise. As a consequence, a realistic use scenario for a ubicomp application includes not only the people, artifacts, and places involved in a single target activity, but potentially 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. In addition, prototypes themselves often have properties that may make them stand out. To be unobtrusive, prototypes work best when they are refined, are of appropriate size and weight, and require only appropriate amounts of attention (such as the ambient display in Figure 6d). This makes evaluation at the early stages of design particularly challenging. Consider the three prototypes shown in Figure 9, 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 one research goal of ubicomp to be systems that integrate into "the practical logic of the routine world" (Anderson, 1994, p. 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 7. Video recordings and system logs are both helpful, but the traditional methods of working with

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Figure 7. **The Plasma Poster is an interactive public display designed to encourage informal content sharing and conversations. The system is designed for informal social situations, such as a café (pictured here), which are difficult to recreate in lab settings (Churchill, Nelson, Denoue, Helfman, & Murphy, 2004). Copyright 2006 by Elizabeth Churchill. Reprinted with permission.**

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

these data have often been prohibitively time consuming. Lighter weight techniques for dealing with rich capture of longitudinal user data are needed.

3. STRATEGIES FOR ECOLOGICALLY VALID DESIGN

All of these issues may make ecologically valid design difficult. Although this can seem daunting—and indeed, the difficulty of these issues may be a central reason for the paucity of evaluation—we suggest that developers have a small but growing set of tools supporting self-report, prototyping, and deployment that can help overcome the challenges of evaluating user behavior in realistic settings. Note that we are not arguing for particular processes, sets of metrics, or intended outcomes here. Instead, we present a set of tools that can be chosen and used according to the needs and goals of a particular developer or researcher.

3.1. Observation

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

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

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 (Hammersley & Atkinson, 1995)—comprising intensive qualitative observation that can last multiple years—to more cost-sensitive and applied methods such as contextual design (Beyer & Holtzblatt, 1998; Holtzblatt, Wendell, & Wood, 2005). Returning to our working definition of ubicomp as being computing that is concerned with "the practical logic of the routine world" (Anderson, 1994, p. 178), it becomes clear why qualitative field observation methods have enjoyed some success in user-centered ubiquitous computing efforts (see e.g., Consolvo et al., 2005; Grinter & Eldridge, 2001; Hulkko, Mattelmaki, Virtanen, & Keinonen, 2004; Okabe & Ito, 2006; Palen et al., 2000).

The primary difficulty with gathering high-quality data through observation is remaining *unobtrusive* while monitoring potentially *sparse data.* Lower cost observational methods that are perfectly appropriate for more constrained settings may be less successful at handling unobtrusiveness and sparse data. Although a carefully structured evaluation can help to mitigate this, evaluators may be forced to reduce realism in the process (e.g., by simulating events at a higher frequency than they might otherwise happen in order to observe a participant's response).

When realism is important, one may turn to situated techniques that allow for a remote evaluator. This can make it feasible to conduct evaluations over a longer period (addressing data sparsity). Also, the removal of the evaluator to a remote location to make the experiment less obtrusive. Of course monitoring can still interfere as long as the user is involved or aware of data being gathered. Another challenge in observation efforts is that capturing data is often cheap and easy but accessing that data later for use as a design resource can be challenging. Interfaces that help manage these 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 (Yeh et al., 2006). Next we discuss two particular situated techniques that are especially appropriate for ubicomp because they can provide a balanced solution to the problems of realism, unobtrusiveness, and data sparsity.

Experience Sampling

In the Experience Sampling Method (ESM), participants are interrupted throughout the day to answer a set of questions at predefined (or random) intervals specified by the researcher. Participants typically must respond to a short survey. The technique in its classical form is very appropriate for the needs of ubicomp.

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

By asking questions at a low frequency, and keeping the experimenter remote, the technique can remain fairly unobtrusive. 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 Intille, Kukla, & Ma, 2002; Rondini, 2003; Wheeler & Rois, 1991, Iachello, et al., 2006; for more information on this technique, which is illustrated in Figure 8). Ideally, event-contingent ESM asks questions only at the rare moments when something interesting happens, rather than hoping that question and event will coincide.

Although 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 Beaudin, Intille, & Tapia, 2004).

Diary Studies

One problem with ESM is that when researchers control capture, they are able to obtain objective data about participants' activities but do not necessarily gain an understanding of the events that are important to the participants. The diary study is a method of understanding participant behavior and intent in situ in which participants control the timing and means of capture. Participants in a diary study are typically told to watch for certain critical events (e.g., "Write down moments that involve searching for, consuming, or producing information"). One drawback of diary studies is that events important to researchers may not be important to, and therefore not captured by, participants—a problem that a hybrid ESM/diary study approach can address.

Figure 8. **A mobile PDA interface for event-contingent experience sampling. The interface shown here is from the Context-Aware Experience Sampling Tool (Intille, Kukla, & Ma, 2002). Copyright 2006 by MIT Stephen S. Intille. Reprinted with permission.**

Diary studies can handle data sparsity by specifically instructing participants to report events rather than relying on luck or sensing as ESM does. Recent work also pays special attention to realism. Today's digital devices make it feasible for participants to capture a variety of media along with their own handwritten thoughts or answers to questions. Captured media can be quite rich and diverse, and having participants discuss artifacts can be a powerful data-gathering technique. Carter and Mankoff (2005b) compared the impact of different media on diary studies and identified the varied contributions of timing information, event sequencing, audio, and video to activity reconstruction.

Because of their reliance on participants to decide what to record, diary studies are not as well suited as ESM to a broad sample of all of a day's events. However, methods that combine ESM and diaries (such as the day reconstruction method as described in Kahneman, Krueger, Schkade, Schwarz, & Stone, 2004) may provide both breadth (some information about all events) as well as depth (details about important events).

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

3.2. Lightweight Prototyping and Iteration

Although 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 (T. Kelley, 2001). 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 (Schön & Bennett, 1996)—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. They provide artifacts that can be used for user feedback and usability testing. Fourth, prototypes are 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, which simulate "the concrete sensory experience of using an artifact" (Houde & Hill, 1997, p. 3) and show only the form of a device, such as Figure 9 (left and middle), and *works-like* prototypes, which use a computer display to demonstrate functionality and more closely simulate actual user experience (Buchenau & Suri, 2000), such as Figure 9 (right). The time and expertise requirements for cre-

Figure 9. **A paper sketch, physical mock-up, and final prototype (implemented with d.tools), showing how the interface of Klemmer, Hartmann, and Takayama's (2006) SnuzieQ, an alarm clock, evolved through prototyping. This figure is reprinted from "How Bodies Matter: Five Themes for Interaction Design," by S. R. Klemmer, B. Hartmann, and L. Takayama, 2006, *Proceedings of the DIS 2006 Symposium on Designing Interactive Systems,* 142. Copyright 2006 ACM, Inc. Reprinted with permission.**

ating 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 (Ulrich & Eppinger, 2000).

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 (Rettig, 1994; Snyder, 2003), probes, and Wizard of Oz simulations of working systems. All of the challenges are problematic at this stage of development. Although 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 (Landay, 1996). However, as we saw in Section 2, it is 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 an experimenter 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 nonsituated settings, they can also provide straightforward ways to explore the impact of ambiguity (a developer

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

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 (Dahlbäck et al., 1993; J. F. Kelley, 1984; Maulsby et al., 1993). 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 been shown to be useful for simulating recognizers (Akers, 2006), multimodal interfaces (Chandler, Lo, & Sinha, 2002; Oviatt et al., 2000), sensing (Consolvo, Roessler, & Shelton, 2004; Hudson et al., 2003; Mynatt, Rowan, Craighill, & Jacobs, 2001), intelligent user interfaces (Dahlbäck, Jönsson & Ahrenberg, 1993), location (Benford et al., 2004; Li, Hong, & Landay, 2004), augmented reality (MacIntyre, Gandy, Dow, & Bolter, 2004), and input technologies (Klemmer, Li, Lin, & Landay, 2004b) early in the design process. Once software is developed, Wizard of Oz–enabled tools can assist in the collection and analysis of usability data and in reproducing scenarios during development and debugging (Klemmer et al., 2000). 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, analyze philosophy utilized in SUEDE (Klemmer et al., 2000), a tool that allows designers to prototype prompt/response speech interfaces, and Momento (Carter, Mankoff, & Heer, 2007b), a tool that supports ubicomp experimentation.

Another approach to achieving realism with works-like prototypes is to create robust prototypes with simple functionality that can be rapidly created and deployed to probe use patterns. The original culture probes introduced by Gaver, Donne, and Pacenti (1999) have been expanded to include technology (Hutchinson et al., 2003; Paulos & Goodman, 2004; Paulos & Jenkins, 2005). 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. (Hutchinson et al., 2003, p. 17)

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

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, such as with paratypes (Abowd et al., 2006).

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-size kitchen, Liu and Khooshabeh (2003) 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, whereas the interactive prototype took two weeks. In a different study, Mankoff and Schilit (1997) deployed paper prototypes of an application (shown in Figure 10) 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.

3.3. 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" (Abowd & Mynatt, 2000a, p. 49).

Eventually, it becomes necessary to deploy a real prototype in the field. These prototypes go beyond the lightweight representations previously mentioned to include real interaction. Although high-fidelity implementation of ubiquitous computing systems deserves a longer discussion than space af-

Figure 10. **A picture of PALplates in use. The top row of stickies (with the pictures on them) indicates functionality (Mankoff & Schilit, 1997). The stickies below each function were placed there and written on by end users. This figure is reprinted from "Prototypes in the Wild: Lessons from Three Ubicomp Systems," by S. Carter and J. Mankoff, 2005, *IEEE Pervasive Computing, 4*(4), 52. Copyright 2005 IEEE. Reprinted with permission.**

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

fords, we highlight a few particularly salient issues here: It is difficult to develop systems robust enough for realistic situations and to recover from breakdowns quickly enough to sustain a critical mass of users.

As Section 2 demonstrated, reasons for lack of iteration include the expertise and the time necessary to build 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. (2006) found that although 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 (Abowd, 1999b). 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 for ubicomp applications being field tested (Abowd, 1999b). Researchers and developers have created some tools and toolkits to allow developers to rapidly prototype ubicomp applications for early-stage testing

(including Dey, Abowd, Salber, 2001; Greenberg & Fitchett, 2001; Klemmer, Li, Lin, & Landay, 2004; Matthews et al., 2004; http://mobile.processing.org/; http://www.forum.nokia.com/python/). 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.

3.4. Controlled Evaluations

Controlled evaluations comprise laboratory experiments, field simulations, and controlled field experiments (McGrath, 1995). 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 re-creating 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, Skov, Als, and Høegh (2004a) found that a laboratory test approximating field use found usability problems at a lower cost than field experiments. Kjeldskov and Stage (2004b) 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 can help to increase the usefulness of controlled evaluations for ubicomp developers.

3.5. Field Experiments

When ubicomp applications are deployed and used (or even commercialized), it gives the field valuable data about what really works or does not work.

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

As previously noted, 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 at the deployment site who can help to speed up acceptance and to increase the chance of success (Carter et al., 2004). 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 (Abowd, 1999a; Abowd et al., 2000b). 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 and colleagues (Intille et al., 2005; Intille et al., 2006) 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 a system deployed in a busy public space like the Plasma Poster (Churchill et al., 2004). 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 (Raento, Oulasvirta, Petit, & Toivonen, 2005), or adapting the needfinding techniques discussed earlier to encourage users to introspect on their situated use of deployed technologies.

Given that it is difficult to evaluate one prototype, it is clearly also challenging to conduct an experiment comparing multiple prospective designs. To address this issue, Trevor, Hilbert, and Shilit (2002) 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. 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 use-

ful" (p. 66). 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 that the different interfaces had stengths and weaknesses that varied with the nature of the task and the context of use.

3.6. Summary

McGrath (1995) 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, whereas 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. In this section, we have shown that developers have a small but growing set of tools to overcome challenges evaluating user behavior with ubicomp applications in realistic settings: self-report methods for needfinding; Wizard of Oz, paper prototyping, and probes for lightweight prototyping; research and professional toolkits for functional prototyping; methods of re-creating environments for controlled evaluations; and a set of approaches to encourage use, gather data, and compare designs in field experiments.

4. IMPLICATIONS

Thus far, we have argued that implementing and evaluating ubicomp systems is difficult and time consuming because of the scale and sensing challenges that ubicomp introduces. In this section, we suggest research directions that could help address issues of sensing and scale, either by easing the path to prototyping and implementation or by enabling researchers to better handle these challenges when conducting evaluations.

4.1. Conversations With Materials

Walking into a design studio, one can see Barbie dolls, umbrellas, new ideas, old ideas, good ideas, and bad (see Figure 11). The abundance of artifacts makes the question "What are you doing?" obsolete. Collocated, cluttered studios are hallmarks of design practice. The physical manifestation of the studio affords peer learning, discussion, and constant critique of work in progress. This "technology" was introduced with the founding of École des Beaux-Arts in Paris in 1819 and has endured for nearly 200 years.

Mundane materials such as cardboard, hot glue, and foam core play a marquee role in contemporary product design. Prototypes, often made from these materials, are the pivotal medium that structures innovation, collaboration,

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Figure 11. **Like many art and design studios, the open-plan architecture and ubiquity of the physical materials of a craft in the Stanford Product Design studio space affords a visibility of work practice—this visibility is notably absent in PC-based spaces such as cubicle farms. This figure is reprinted from "How Bodies Matter: Five Themes for Interaction Design," by S. R. Klemmer, B. Hartmann, and L. Takayama, 2006, *Proceedings of DIS 2006 Symposium on Designing Interactive Systems*, 144. Copyright 2006 ACM, Inc. Reprinted with permission.**

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

and creativity in the most successful design studios. In Schrage's (1999) words, "Organizations manage themselves by managing their prototypes" (p. 61). As we enter the age of ubiquitous computing, what prototyping tools and environments will enable the design of ubicomp devices to be as quick and fluid as foam core and hot glue are for passive objects today?

Currently, the integrated prototyping of bits and atoms for ubiquitous computing devices requires resources and knowledge outside the reach of design generalists. Figure 12 shows two examples of research efforts intended to support integrated prototyping by generalists.

d.tools. Based on interviews with product designers, Hartmann et al. (2006) created d.tools, a system enabling nonprogrammers to create the bits and the atoms of physical user interfaces in concert. d.tools lowers the threshold to prototyping functional physical interfaces through plug-and-play hardware that is closely coupled with a visual authoring environment (see Figure 12, left). With d.tools, designers place physical controllers (e.g., buttons, sliders), sensors (e.g., accelerometers, compasses), and output devices (e.g., LEDs, LCD screens, and speakers) directly onto form prototypes and then author behavior visually in our software workbench. The d.tools library includes an extensible set of smart components that cover a wide range of input and output technologies.

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Figure 12. **Two design tools: *Left*: The d.tools software editor and an interactive prototype built in approximately 3 hr (Hartmann et al., 2006). *Right*: With BOXES, designers can prototype physical forms in minutes and then rapidly add simple functionality to them by connecting thumbtacks to on-screen mouse and keyboard actions (Hudson & Mankoff, 2006).**

Cardboard Boxes. The BOXES (Building Objects for eXploring Executable Sketches) system enables rapid creation of prototypes using cardboard and thumbtacks (Hudson & Mankoff, 2006). Thumbtacks can be used as buttons that cause mouse and keyboard actions onscreen, as specified by the designer and shown in Figure 12, right. Thus, a designer can rapidly create a prototype that can control existing or newly prototyped applications.

d.tools functions by employing a PC as a proxy for embedded processors so that designers can focus on user experience-related tasks rather than implementation-related details. Feedback is handled using library elements such as the LCD screen visible in Figure 12 (bottom left). BOXES can be wireless but currently depends on a PC for feedback to the user. A challenging area for future research is to bring the same flexibility to feedback that is brought to physical form and input in both systems. Projected displays that are able to move with and adjust to a moving prototype represent one potential solution to this problem (preliminary work in this area by J. C. Lee, Hudson, Summet, and Dietz (2005), is a promising first step).

4.2. Prototyping for Evaluation

Traditionally, UI design tools have focused on the *creation* of user interfaces, but the evaluation and subsequent analysis of those interfaces has gotten short shrift. We propose that UI design tools should encompass all three stages. Perhaps the most powerful lesson that interaction designers have learned in the past 2 decades is to *fail early, so one can succeed sooner.* Moving from failure to success requires not only building a prototype but also testing

that prototype with users and then making improvements based on that test data.

A prototype's functionality for testing can be *interactive* when tools support rapid construction; *mocked-up* when tools allow the creation of examples that will later be backed by real code; or implemented via Wizard of Oz. Wizard functionality is especially useful for ubicomp technologies that are challenging to implement (e.g., computer vision recognition) and for cooperatively prototyping new functionality on the fly as an evaluation session unfolds.

Prototyping for evaluation implies requirements that are not always made explicit in prototyping tools, especially when those evaluations will be situated in field settings as we have argued ubicomp evaluations often are. For example, a prototyping tool focused on field evaluation might benefit from features such as logging not only application state but also context (e.g., location, nearby Bluetooth devices), easily downloading new or updated functionality, experimenter or system triggered requests for information from the end user, and piggybacking on existing devices already carried by the user such as her mobile phone. Momento, shown in Figure 13, is a tool that leverages the SMS/MMS network to support these features for interactive and Wizard of Oz prototypes (Carter, Mankoff, & Heer, 2007b).

Although Momento is a tool for exploring works-like prototypes with minimal implementation, toolkits such as Papier-Mâché (Klemmer, Li, Lin, & Landay, 2004b) are more focused on supporting the creation of interactive prototypes. By providing a generic, *evaluation-time* wizard interface, toolkits can enable a wizard to control the state and behavior of any aspect of a complex interactive prototype from "behind a curtain."

One dysfunction of current usability practice is that although it is easy to capture usability data such as video and logs of participant actions, accessing that data is prohibitively time consuming. As Crabtree et al. (2006) point out, evaluation support tools can aid designers by capturing a rich set of time-stamped evaluation data and correlating it with application state. After evaluation, usability data can be aggregated and presented to the designer using information visualization techniques. Designers can use these data to reflect on the state of their prototype. Furthermore, tools that extract metadata can facilitate convenient search, and visualizations of results within the same interaction framework that the design tool employs can allow designers to make immediate changes based on the data.

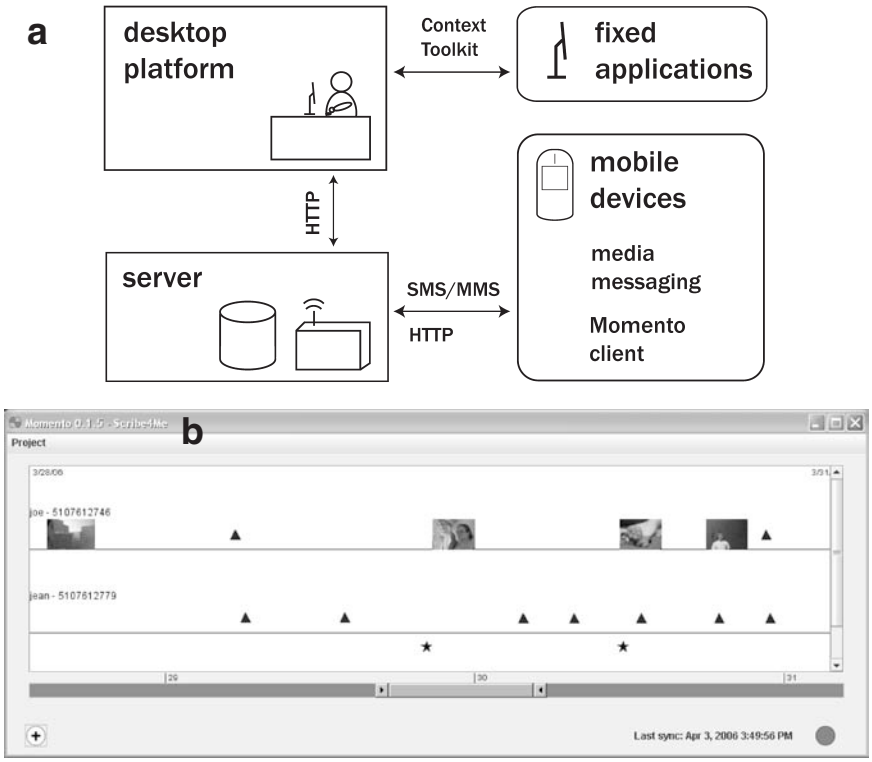
4.3. Supporting In-the-World Evaluation

Wizards and foam core are fantastic techniques for early exploration, but eventually wizards get hungry and foam core wears out. How might future tools and methods research help when it is time to move toward in-the-world

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Figure 13. **The Momento system (Carter, Mankoff, & Heer, 2007b). (a) The desktop platform communicates with mobile devices via a server and with third-party applications using the Context Toolkit. (b) The desktop platform includes a timeline that visualizes events as they are received (triangles) and sent (other shapes). Figures 13a and 13b reprinted from "Momento: Support for Situated Ubicomp Experimentation," by S. Carter, J. Mankoff, and J. Heer, 2007, *Proceedings of the CHI 2007 Conference on Human Factors in Computing Systems*, p. 126, 128. Copyright 2007 ACM, Inc. Reprinted with permission.**



evaluation? We see the following three goals as the most pressing: making it easier to develop robust prototypes, minimizing deployment costs, and minimizing per-participant costs. We use the term *cost* broadly, including monetary, labor, and frustration on the part of the experimenters and the participants.

Much has been written about the research challenges involved in creating ubicomp infrastructure (e.g., Bellotti et al., 2002; Dey, Abowd, & Salber, 2001; Edwards, Bellotti, Dey, & Newman, 2003; Edwards & Grinter, 2001; Grimm et al., 2004; Hong & Landay, 2001; Johanson, Winograd, & Fox, 2003), so we do not recount the full discussion here. For a large majority of ubicomp applications, tremendous resources, expertise, and time must be committed to creating prototypes because of sensing and scale-related

challenges—especially when working with non-PC hardware, from mobile phones to mechatronics. To date, little work has explored how computer science research might support ubicomp prototyping, evaluation, and iteration.

Several challenges for prototyping research also arise from the heterogeneity of ubicomp technologies. First, research could improve the methods by which members of a design team collaborate through design tools. Tools could aid conversations by affording designers some understanding of the technical constraints of a system and technologists an understanding of the user needs, without requiring that either be an expert in the other's domain. Heterogeneous ubicomp technologies also make it challenging to limit the size of a toolkit's library components. With graphical user interfaces, there is a standard set of widgets, and these widgets span nearly all common applications. However, how to limit the library size of ubicomp support tools is an open question. Finally, the heterogeneity of ubicomp technologies would benefit from continued research on model-based design techniques (Szekely, 1996). This would benefit both designers' abilities to explore alternatives and work iteratively and their ability to create interfaces that can be customized for individual situations and user needs.

From a prototyping perspective, the "conversation with materials" that occurs through longitudinal deployment is a valuable one. We encourage tools that more richly support a design-test-analysis approach in the context of longitudinal deployments (Hartmann et al., 2006; Klemmer et al., 2000). Tools that support the capture and mutual presentation of environmental context and user interaction logs are particularly promising, as are systems that benefit from Wizard of Oz support but are careful to respect the wizard's time so that their support is elicited only when it is essential. Momento is a first step in this direction. It provides wizards with a peripheral display of incoming events shown in Figure 13b, and a rules system for handling the more straightforward requests.

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

4.4. Support for Machine Learning and Sensor-Based Interaction

Given the prominence of sensing in ubiquitous computing, it is not surprising that machine learning is beginning to receive increasing attention. Although machine learning is often seen as the domain of non-HCI specialists, it is starting to become clear that HCI techniques can help to identify the best ways to approach problems that involve machine learning (e.g. Fogarty et al., 2005; Rowan & Mynatt, 2005). At the same time, tools that make machine learning more accessible to practitioners are beginning to appear (e.g. Dey, Hamid, Beckmann, Li, & Hsu, 2004; Olsen, Taufer, & Fails, 2004; Witten & Frank, 2005).

Both prototyping tools and evaluation support that addresses issues pertinent to machine learning are needed. As an example, a researcher might wish

to created a rough prototype of a gesture recognizer, use it to Wizard of Oz a study to determine more appropriate gestures, and then use the data from that study to train a more complete recognizer (e.g., Akers, 2006; Long, Landay, & Rowe, 1999). Researchers might also wish to understand the impact of recognition errors and mediation techniques for allowing the end user to manage errors. A prototyping tool that integrates end-to-end support for creating such applications, mediation techniques, and the machine learning systems underlying them is sorely needed.

4.5. Data Sparsity

Situated studies are crucial, and field deployments are a big part of this. One challenge that is not directly addressed by past work is dealing with data sparsity. In field evaluations, this challenge can be addressed by either lengthening an evaluation or including more participants. Tools that help the developer to gather information about particularly important events will help to focus the effort of evaluation where it matters most.

By automating and simplifying some aspects of data collection, Momento facilitates larger and longer evaluations earlier in the design cycle (Carter, Mankoff, & Heer, 2007b). However, when wizards are needed, more participants/time means more wizards. The problem of coordinating multiple wizards simultaneously or over time is still an open challenge.

Finally, highly instrumented environments (e.g., Intille et al. 2006; Kidd et al., 1999) can help with the identification of informative events. However, identifying important events is an open and difficult problem; importance of an event varies with the application, and possibly the user, being studied.

5. CONCLUSIONS

To the extent that, as Schön wrote, prototypes are "reflective conversation[s] with materials" (Schön & Bennett, 1996, p. 171) ubicomp as practiced today is a soliloquy. It shares with us a perspective, a viewpoint, but—broadly speaking—the research community has not checked whether anyone is listening. In this article, through a literature survey and interviews with 28 developers from three ubicomp subfields, we have illustrated how challenges of sensing and scale cause ubicomp systems to resist ecologically valid evaluation. To date, few have addressed how computer science research might support ubicomp prototyping, evaluation, and iteration. To be sure, development is a central piece of that, but this article suggests that development support is a means, not an end in itself.

However, as we have noted, all is not dark. Success stories exist in every aspect of iterative design, and researchers are working hard to develop support-

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

ive tools and techniques. In addition, new approaches are beginning to emerge but need more investigation. For example, sensor reliability data can be incorporated into decision making to help make formative evaluations that involve event recognition and prompts, such as ESM, more unobtrusive (Antifakos, Schwaninger, & Schiele, 2004). Work showing that simplified reconstructions of some field environments produce data at least as reliable as that gathered in the field could mitigate data sparsity for certain situations (Kjeldskov & Stage, 2004a).

Despite this, many open problems exist. Looking forward, we believe that tools and methods that support the sensing and scaling of ubicomp systems will enable researchers to tackle more ecologically valid design. In particular, we see fast prototyping at the intersection of materials (atoms) and interaction (bits), design for evaluation, end-to-end support for machine learning, and methodological triangulation as particularly promising avenues for future development.

Although the "selfish" reason for ecologically valid design is creating systems that solve a problem the designer cares about, the community as a whole benefits from evaluations that provide generalizable results, either as a side effect or as their main goal. Perhaps the hardest challenge left open by existing work on evaluation techniques is creating the possibility of generalizing the results of an evaluation. There is no panacea for this, although triangulation of multiple methods can help. We can hope that the more evaluations that are successfully run, and the more we learn about both similar and differing systems, the better we are able to judge what can and cannot generalize.

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

NOTES

Acknowledgments. This article could not have happened without the insight of many people. We thank Jan Chong, Björn Hartmann, Scott Hudson, and Tye Rattenbury for insightful research discussions and feedback on drafts. We also thank Gregory Abowd, whose 1997 Qualifying Exam question inspired the very first draft of this article, and Anind Dey, whose comments have influenced many of the issues raised here.

Support. This work was funded in part by National Science Foundation grants IIS-0534662, IIS-0205644, and IIS-0209213.

Authors' Present Addresses. Scott Carter, FX Palo Alto Laboratory, Inc., 3400 Hillview Avenue, Building 4, Palo Alto, CA 94304. E-mail: carter@fxpal.com. Jennifer Mankoff, HCI Institute, Carnegie Mellon University, Pittsburgh, PA 15213. E-mail: jmankoff@acm.org. Scott R. Klemmer, HCI Group, Computer Science Department, Stanford University, Palo Alto, CA 94305. E-mail: srk@cs.stanford.edu. Tara Matthews, IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120. E-mail: tlmatthe@us.ibm.com.

HCI Editorial Record. First manuscript received April 24, 2006. Revision received November 14, 2006. Accepted by Jonathan Grudin. Final manuscript received January 30, 2007. —*Editor*

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

References

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

Abowd, G. D. (1999b). Software engineering issues for ubiquitous computing. *Proceedings of the ICSE '99 International Conference on Software Engineering*, 75–84. New York: ACM.

Abowd, G. D., & Mynatt, E. D. (2000a). Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction, 7*(1), 29–58.

Abowd, G. D., Atkeson, C. G., Bobick, A. F., Essa, I. A., MacIntyre, B., Mynatt, E. D., et al. (2000b). Living laboratories: The Future Computing Environments Group at the Georgia Institute of Technology. *Proceedings of the CHI 2000 Conference on Human Factors in Computing Systems, Extended Abstracts.* New York: ACM.

Abowd, G. D., Hayes, G. R., Iachello, G., Kientz, J. A., Patel, S. N., & Stevens, M. M. (2005). Prototypes and paratypes: Designing mobile and ubiquitous computing applications. *IEEE Pervasive Computing, 4*(4), 67–73.

Akers, D. (2006). Cinch: A cooperatively designed marking interface for 3D pathway selection. *Proceedings of the UIST 2006 Symposium on User Interface Software and Technology.* New York: ACM.

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

Antifakos, S., Schwaninger, A., & Schiele, B. (2004). Evaluating the effects of displaying uncertainty in context-aware applications. *Proceedings of the Ubicomp 2004 International Conference on Ubiquitous Computing.* Berlin: Springer.

Ballagas, R., Ringel, M., Stone, M., & Borchers, J. (2003). iStuff: A physical user interface toolkit for ubiquitous computing environments. *Proceedings of the CHI 2003 Conference on Human Factors in Computing Systems, Extended Abstracts.* New York: ACM.

Beaudin, J., Intille, S., & Tapia, E. M. (2004). Lessons learned using ubiquitous sensors for data collection in real homes. *Proceedings of the CHI 2004 Conference on Human Factors in Computing Systems, Extended Abstracts.* New York: ACM.

Bellotti, V., Back, M., Edwards, W. K., Grinter, R. E., Henderson, A., & Lopes, C. V. (2002). Making sense of sensing systems: Five questions for designers and researchers. *Proceedings of the CHI 2002 Conference on Human Factors in Computing Systems.* New York: ACM.

Benford, S., Crabtree, A., Flintham, M., Drozd, A., Anastasi, R., Paxton, M., et al. (2006). Can you see me now? *ACM Transactions on Computer-Human Interaction, 13*(1), 100–133.

Benford, S., Seager, W., Flintham, M., Anastasi, R., Rowland, D., Humble, J., et al. (2004). The error of our ways: The experience of self-reported position in a location-based game. *Proceedings of Ubicomp 2004 International Conference on Ubiquitous Computing.* Berlin: Springer.

Beyer, H., & Holtzblatt, K. (1998). *Contextual design: Defining customer-centered systems (Series in interactive technologies).* San Francisco: Morgan Kaufmann.

Buchenau, M., & Suri, J. F. (2000). Experience prototyping. *Proceedings of DIS 2000 Symposium on Designing Interactive Systems.* New York: ACM.

Cadiz, J., Venolia, G., Jancke, G., & Gupta, A. (2002). Designing and deploying an information awareness interface. *Proceedings of CSCW 2002 Conference on Computer Supported Cooperative Work.* New York: ACM.

Carter, S., Mankoff, J., & Goddi, P. (2004). Building connections among loosely coupled groups: Hebb's rule at work. *Journal of Computer Supported Cooperative Work, 13*(3), 305–327.

Carter, S., & Mankoff, J. (2005a). Prototypes in the wild: Lessons learned from evaluating three ubicomp systems. *IEEE Pervasive Computing, 4*(4), 51–57.

Carter, S., & Mankoff, J. (2005b). When participants do the capturing: The role of media in diary studies. *Proceedings of CHI 2005 Conference on Human Factors in Computing Systems.* New York: ACM.

Carter, S. (2007a). *Supporting early-stage ubicomp experimentation.* PhD diss., University of California, Berkeley.

Carter, S., Mankoff, J., & Heer, J. (2007b). Supporting needs gathering and prototyping of situation dependent ubicomp applications with Momento. *Proceedings of the CHI 2007 Conference on Human Factors in Computing Systems.* New York: ACM.

Chandler, C. D., Lo, G., & Sinha, A. K. (2002). Multimodal theater: Extending low fidelity paper prototyping to multimodal applications. *Proceedings of CHI 2002 Conference on Human Factors in Computing Systems, Extended Abstracts.* New York: ACM.

Churchill, E. F., Nelson, L., Denoue, L., Helfman, J., & Murphy, P. (2004). Sharing multimedia content with interactive public displays: A case study. *Proceedings of DIS 2004 Symposium on Designing Interactive Systems.* New York: ACM.

Consolvo, S., Roessler, P., & Shelton, B. E. (2004). The CareNet display: Lessons learned from an in home evaluation of an ambient display. *Proceedings of Ubicomp 2004 International Conference on Ubiquitous Computing.* Berlin: Springer.

Consolvo, S., Smith, I. E., Matthews, T., LaMarca, A., Tabert, J., & Powledge, P. (2005). Location disclosure to social relations: Why, when, and what people want to share. *Proceedings of CHI 2005 Conference on Human Factors in Computing Systems.* New York: ACM.

Crabtree, A., Benford, S., Greenhalgh, C., Tennent, P., Chalmers, M., & Brown, B. (2006). Supporting ethnographic studies of ubiquitous computing in the wild. *Proceedings of DIS 2006 Symposium on Designing Interactive Systems.* New York: ACM.

Dahlbäck, N., Jönsson, A., & Ahrenberg, L. (1993). Wizard of Oz studies—Why and how. *Proceedings of IUI '93 International Conference on Intelligent User Interfaces.* New York: ACM.

Détienne, F. (2001). *Software design–cognitive aspects (Practitioner series).* London: Springer-Verlag.

Dey, A. K., Abowd, G. D., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction, 16*(2–4), 97–166.

Dey, A. K., Hamid, R., Beckmann, C., Li, I., & Hsu, D. (2004). *a CAPpella*: Programming by demonstration of context-aware applications. *Proceedings of CHI 2004 Conference on Human Factors in Computing Systems.* New York: ACM.

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Dey, A. K., & Mankoff, J. (2005). Designing mediation for context-aware applications. *ACM Transactions on Computer-Human Interaction, 12*(1), 53–80.

Dourish, P. (2001). *Where the action is: The foundations of embodied interaction.* Cambridge, MA: MIT Press.

Edwards, W. K., Bellotti, V., Dey, A. K., & Newman, M. W. (2003). The challenges of user-centered design and evaluation for infrastructure. *Proceedings of the CHI 2003 Conference on Human Factors in Computing Systems.* New York: ACM.

Edwards, W. K., & Grinter, R. E. (2001). At home with ubiquitous computing: Seven challenges. *Proceedings of the Ubicomp 2001 International Conference on Ubiquitous Computing.* Berlin: Springer.

Fitzmaurice, G. W., Ishii, H., & Buxton, W. (1995). Bricks: Laying the foundations for graspable user interfaces. *Proceedings of the CHI '95 Conference on Human Factors in Computing Systems.* New York: ACM.

Fleck, M., Frid, M., Kindberg, T., O'Brien-Strain, E., Rajani, R., & Spasojevic, M. (2002). Rememberer: A tool for capturing museum visits. *Proceedings of the Ubicomp 2002 International Conference on Ubiquitous Computing.* Berlin: Springer.

Fogarty, J., Hudson, S. E., Atkeson, C. G., Avrahami, D., Forlizzi, J., Kiesler, S., et al. (2005). Predicting human interruptibility with sensors. *ACM Transactions on Computer-Human Interaction, 12*(1), 119–146.

Gaver, B., Dunne, T., & Pacenti, E. (1999). Design: Cultural probes. *Interactions, 6*(1), 21–29.

Gonzalez, V. M., & Mark, G. (2004). "Constant, constant, multi-tasking craziness": Managing multiple working spheres. *Proceedings of the CHI 2004 Conference on Human Factors in Computing Systems.* New York: ACM.

Greenberg, S., & Fitchett, C. (2001). Phidgets: Easy development of physical interfaces through physical widgets. *Proceedings of the UIST 2001 Symposium on User Interface Software and Technology.* New York: ACM.

Grimm, R., Davis, J., Lemar, E., Macbeth, A., Swanson, S., Anderson, T., et al. (2004). System support for pervasive applications. *ACM Transactions on Computer Systems, 22*(4), 421–486.

Grinter, R. E., & Eldridge, M. (2001). y do tngrs luv 2 txt msg? *Proceedings of the ECSCW 2001 European Conference on Computer Supported Cooperative Work.* Berlin: Springer.

Grønbæk, K., Kyng, M., & Mogensen, P. (1992). CSCW challenges in large-scale technical projects—A case study. *Proceedings of the CSCW '92 Conference on Computer Supported Cooperative Work.* New York: ACM.

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

Hammersley, M., & Atkinson, P. (1995). *Ethnography: Principles in practice.* London: Routledge.

Hartmann, B., Klemmer, S. R., Bernstein, M., Abdulla, L., Burr, B., Robinson-Mosher, A., et al. (2006). Reflective physical prototyping through integrated design, test, and analysis. *Proceedings of the UIST 2006 Symposium on User Interface Software and Technology.* New York: ACM.

Heiner, J. M., Hudson, S. E., & Tanaka, K. (1999). The information percolator: Ambient information display in a decorative object. *Proceedings of the UIST '99 Symposium on User Interface Software and Technology.* New York: ACM.

Herbsleb, J., Atkins, D. Boyer, D., Handel, M., & Finholt, T. (2002). Introducing instant messaging and chat in the workplace. *Proceedings of the CHI 2002 Conference on Human Factors in Computing Systems.* New York: ACM.

Hill, R. D. (1986). Supporting concurrency, communication and synchronization in human-computer interaction—The Sassafras UIMS. *ACM Transactions on Graphics, 5*(3), 179–210.

Ho-Ching, W.-L., Mankoff, J., & Landay, J. (2003). Can you see what I hear? The design and evaluation of a peripheral sound display for the deaf. *Proceedings of the CHI 2003 Conference on Human Factors in Computing Systems.* New York: ACM.

Holmquist, L. E. (2005). Prototyping: generating ideas or cargo cult designs? *Interactions, 12*(2), 48–54.

Holtzblatt, K., Wendell, J. B., & Wood, S. (2005). *Rapid contextual design: A how-to guide to key techniques for user-centered design.* San Francisco: Morgan Kaufmann.

Hong, J. I., & Landay, J. A. (2001). An infrastructure approach to context-aware computing. *Human-Computer Interaction, 16*(2–4), 287–303.

Horst, H., & Miller, D. (2005). From kinship to link-up: Cell phones and social networking in Jamaica. *Current Anthropology, 46*(5), 755–778.

Houde, S., & Hill, C. (1997). What do prototypes prototype? In M. Helander, T. E. Landauer, & P. Prabhu (Eds.), *Handbook of human-computer interaction* (2nd ed.) (pp. 367–381). Amsterdam: Elsevier Science.

Hudson, S. E., Fogarty, J., Atkeson, C. G., Forlizzi, J., Kielser, S., Lee, J. C., et al. (2003). Predicting human interruptibility with sensors: A Wizard of Oz feasibility study. *Proceedings of the CHI 2003 Conference on Human Factors in Computing Systems.* New York: ACM.

Hudson, S., & Mankoff, J. (2006). Rapid construction of functioning physical interfaces from cardboard, thumbtacks and masking tape. *Proceedings of the UIST 2006 Symposium on User Interface Software and Technology.* New York: ACM.

Hulkko, S., Mattelmäki, T., Virtanen, K., & Keinonen, T. (2004). Mobile probes. *Proceedings of NordiCHI 2004 Nordic Conference on Human-Computer Interaction.* New York: ACM.

Hutchinson, H., Mackay, W., Westerlund, B., Bederson, B. B., Druin, A., Plaisant, C., et al. (2003). Technology probes: Inspiring design for and with families. *Proceedings of the CHI 2003 Conference on Human Factors in Computing Systems.* New York: ACM.

Iachello, G., Truong, K., Abowd, G., Hayes, G., & Stevens, M. (2006). Prototyping and sampling experience to evaluate ubiquitous computing privacy in the real world. *Proceedings of the CHI 2006 Conference on Human Factors in Computing Systems.* New York: ACM.

Intille, S., Kukla, C., & Ma, X. (2002). Eliciting user preferences using image-based experience sampling and reflection. *Proceedings of the CHI 2002 Conference on Human Factors in Computing Systems, Extended Abstracts.* New York: ACM.

Intille, S. S., Larson, K., Beaudin, J. S., Nawyn, J., Tapia, E. M., & Kaushik, P. (2005). A living laboratory for the design and evaluation of ubiquitous computing technologies. *Proceedings of the CHI 2005 Conference on Human Factors in Computing Systems, Extended Abstracts.* New York: ACM.

Intille, S. S., Larson, K., Tapia, E. M., Beaudin, J., Kaushik, P., Nawyn, J., et al. (2006). Using a live-in laboratory for ubiquitous computing research. *Proceedings of the Pervasive 2006 International Conference on Pervasive Computing.* Berlin: Springer.

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Ishii, H., Ren, S., & Frei, P. (2001). Pinwheels: Visualizing information flow in architectural space. *Proceedings of the CHI 2001 Conference on Human Factors in Computing Systems, Extended Abstracts.* New York: ACM.

Johanson, B., Winograd, T., & Fox, A. (2003). Interactive workspaces. *IEEE Computer, 36*(4), 99–101.

Kahneman, D., Krueger, A., Schkade, D., Schwarz, N., & Stone, A. (2004). A survey method for characterizing daily life experience: The day reconstruction method. *Science, 306*(5702), 1776–1780.

Kelley, J. F. (1984). An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Office Information Systems, 2*(1), 26–41.

Kelley, T. (2001). *The art of innovation.* New York: Doubleday.

Kidd, C. D., Orr, R., Abowd, G. D., Atkeson, C. G., Essa, I. A., MacIntyre, B., et al. (1999). The Aware Home: A living laboratory for ubiquitous computing research. *Proceedings of the CoBuild '99 Cooperative Buildings, Integrating Information, Organization, and Architecture, Second International Workshop.* Berlin: Springer.

Kjeldskov, J., & Graham, C. (2003). A review of mobile HCI research methods. *Proceedings of the Mobile HCI 2003 International Conference on Human Computer Interaction with Mobile Devices and Services.* Berlin: Springer.

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

Kjeldskov, J., Skov, M. B., Als, B. S., & Høegh, R. T. (2004a). Is it worth the hassle? Exploring the added value of evaluating the usability of context-aware mobile systems in the field. *Proceedings of the MobileHCI 2004 International Conference on Human Computer Interaction with Mobile Devices and Services.* Berlin: Springer.

Kjeldskov, J., & Stage, J. (2004b). New techniques for usability evaluation of mobile systems. *International Journal of Human-Computer Studies, 60*(5–6), 599–620.

Klemmer, S. R. (2004a). *Tangible user interfaces: Tools and techniques.* Unpublished doctoral dissertation, University of California, Berkeley.

Klemmer, S. R., Hartmann, B., & Takayama, L. (2006). How bodies matter: Five themes for interaction design. *Proceedings of the DIS 2006 Symposium on Designing Interactive Systems.* New York: ACM.

Klemmer, S. R., Li, J., Lin, J., & Landay, L. (2004b). Papier-mâché: Toolkit support for tangible input. *Proceedings of the CHI 2004 Conference on Human Factors in Computing Systems.* New York: ACM.

Klemmer, S. R., Newman, M. W., Farrell, R., Bilezikjian, M., & Landay, J. A. (2001). The Designers' Outpost: A tangible interface for collaborative web site design. *Proceedings of the UIST 2001 Symposium on User Interface Software and Technology.* New York: ACM.

Klemmer, S. R., Sinha, A. K., Chen, J., Landay, J. A., Aboobaker, N., & Wang, A. (2000). SUEDE: A Wizard of Oz prototyping tool for speech user interfaces. *Proceedings of the UIST 2000 Symposium on User Interface Software and Technology.* New York: ACM.

Landay, J. A. (1996). *Interactive sketching for the early stages of user interface design.* Unpublished doctoral dissertation, Carnegie Mellon University, Pittsburgh, PA.

Lee, J. C., Hudson, S. E., Summet, J. W., & Dietz, P. H. (2005). Moveable interactive projected displays using projector based tracking. *Proceedings of the UIST 2005 Symposium on User Interface Software and Technology.* New York: ACM.

Li, Y., Hong, J. I., & Landay, J. A. (2004). Topiary: A tool for prototyping location-enhanced applications. *Proceedings of the UIST 2004 Symposium on User Interface Software and Technology.* New York: ACM.

Liu, L., & Khooshabeh, P. (2003). Paper or interactive? A study of prototyping techniques for ubiquitous computing environments. *Proceedings of the CHI 2003 Conference on Human Factors in Computing Systems, Extended Abstracts.* New York: ACM.

Long, A. C., Jr., Landay, J. A., & Rowe, L. A. (1999). Implications for a gesture design tool. *Proceedings of the CHI '99 Conference on Human Factors in Computing Systems.* New York: ACM.

MacIntyre, B., Gandy, M., Dow, S., & Bolter, J. D. (2004). DART: A toolkit for rapid design exploration of augmented reality experiences. *Proceedings of the UIST 2004 Symposium on User Interface Software and Technology.* New York: ACM.

Mackay, W. E. (1998a). Triangulation within and across HCI disciplines. *Human-Computer Interaction, 13*(3), 310–315.

Mackay, W. E., Fayard, A.-L., Frobert, L., & Médini, L. (1998b). Reinventing the familiar: Exploring an augmented reality design space for air traffic control. *Proceedings of the CHI '98 Conference on Human Factors in Computing Systems.* New York: ACM.

Maldonado, H., Lee, B., Klemmer, S. R. & Pea, R. D. (2007). Patterns of collaboration in design courses: Team dynamics affect technology appropriation, artifact creation, and course performance. *Proceedings of the CSCL International Computer Supported Collaborative Learning Conference.* Mahwah, NJ: Lawrence Erlbaum Associates.

Mankoff, J., Dey, A. K., Hsieh, G., Kientz, J., Ames, M., & Lederer, S. (2003). Heuristic evaluation of ambient displays. *Proceedings of the CHI 2003 Conference on Human Factors in Computing Systems.* New York: ACM.

Mankoff, J., Hudson, S. E., & Abowd, G. D. (2000). Interaction techniques for ambiguity resolution in recognition-based interfaces. *Proceedings of the UIST 2000 Symposium on User Interface Software and Technology.* New York: ACM.

Mankoff, J., & Schilit, B. (1997). Supporting knowledge workers beyond the desktop with PALPlates. *Proceedings of the CHI '97 Conference on Human Factors in Computing Systems, Extended Abstracts.* New York: ACM.

Matthews, T. (2007a). *Designing and evaluating glanceable peripheral visualizations.* PhD diss., University of California, Berkeley

Matthews, T., Carter, S., Fong, J., Pai, C., & Mankoff, J. (2006a). Scribe4me: Evaluating a mobile sound translation tool for the deaf. *Proceedings of the Ubicomp 2006 International Conference on Ubiquitous Computing.* Berlin: Springer.

Matthews, T., Czerwinski, M., Robertson, G., & Tan, D. (2006b). Clipping lists and change borders: Improving multitasking efficiency with peripheral information design. *Proceedings of the CHI 2006 Conference on Human Factors in Computing Systems.* New York: ACM.

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

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

Matthews, T., Rattenbury, T., & Carter, S. (2007b). Defining, designing, and evaluating peripheral displays: An analysis using activity theory. *Human-Computer Interaction, 22*(1–2), 221–261.

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Maulsby, D., Greenberg, S., & Mander, R. (1993). Prototyping an intelligent agent through Wizard of Oz. *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems.* New York: ACM.

McCrickard, D. S., Chewar, C. M., Somervell, S., & Ndiwalana, A. (2003). Composing paper and tangible, multiA model for notification systems evaluation—Assessing user goals for multitasking activity. *ACM Transactions on Computer-Human Interaction, 10*(4), 312–338.

McGee, D. R., Cohen, P. R., Wesson, R. M., & Horman, S. (2002). Comparing paper and tangible, multimodal tools. *Proceedings of the CHI 2002 Conference on Human Factors in Computing Systems.* New York: ACM.

McGrath, J. E. (1995). Methodology matters: Doing research in the behavioral and social sciences. In R. M. Baecker, J. Grudin, & W. A. S. Buxton (Eds.), *Readings in Human-Computer Interaction: Toward the Year 2000* (2nd ed., pp. 152–169). San Francisco: Morgan Kaufman.

Moran, T. P., Chiu, P., & van Melle, W. (1997). Pen-based interaction techniques for organizing material on an electronic whiteboard. *Proceedings of the UIST '97 Symposium on User Interface Software and Technology.* New York: ACM.

Mynatt, E., Rowan, J., Craighill, S., & Jacobs, A. (2001). Digital family portraits: Providing peace of mind for extended family members. *Proceedings of the CHI 2001 Conference on Human Factors in Computing Systems.* New York: ACM.

Newman, M. W., Lin, J., Hong, J. I., & Landay, J. A. (2003). DENIM: An informal web site design tool inspired by observations of practice. *Human-Computer Interaction, 18*(3), 259–324.

Nielsen, J. (1989). Usability engineering at a discount. In G. Salvendy & M. Smith (Eds.), *Designing and using human-computer interfaces and knowledge based systems* (pp. 394–401). Amsterdam: Elsevier Science.

Okabe, D., & Ito, M. (2006). Everyday contexts of camera phone use: Steps toward technosocial ethnographic frameworks. In J. Höflich & M. Harmann (Eds.), *Mobile communication in everyday life: An ethnographic view* (pp. 79–102). Berlin: Frank and Timme.

Olsen, D. R., Taufer, T., & Fails, J. A. (2004). ScreenCrayons: Annotating anything. *Proceedings of the UIST 2004 Symposium on User Interface Software and Technology.* New York: ACM.

Olson, G. M., & Olson, J. S. (2000). Distance matters. *Human-Computer Interaction, 15*(2–3), 139–178.

Oulasvirta, A., Tamminen, S., Roto, V., & Kuorelahti, J. (2005). Interaction in 4-second bursts: The fragmented nature of attentional resources in mobile HCI. *Proceedings of the CHI 2005 Conference on Human Factors in Computing Systems.* New York: ACM.

Oviatt, S., Cohen, P., Wu, L., Duncan, L., Suhm, B., Bers, J., et al. (2000). 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.

Palen, L., Salzman, M. C., & Youngs, E. (2000). Going wireless: Behavior & practice of new mobile phone users. *Proceedings of the CSCW 2000 Conference on Computer Supported Cooperative Work.* New York: ACM.

Paulos, E., & Goodman, E. (2004). The familiar stranger: Anxiety, comfort, and play in public places. *Proceedings of the CHI 2004 Conference on Human Factors in Computing Systems.* New York: ACM.

Paulos, E., & Jenkins, T. (2005). Urban probes: Encountering our emerging urban atmospheres. *Proceedings of the CHI 2005 Conference on Human Factors in Computing Systems.* New York: ACM.

Raento, M., Oulasvirta, A., Petit, R., & Toivonen, H. (2005). ContextPhone —A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing, 4*(2), 51–59.

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

Rondini, J. C. (2003). *Context-aware experience sampling for the design and study of ubiquitous technologies.* Unpublished master's thesis, EECS Department, Massachusetts Institute of Technology, Cambridge, MA.

Rowan, J., & Mynatt, E. D. (2005). Digital family portrait field trial: Support for aging in place. *Proceedings of the CHI 2005 Conference on Human Factors in Computing Systems.* New York: ACM.

Rubin, J. (1994). *Handbook of usability testing: How to plan, design, and conduct effective tests.* New York: Wiley.

Rudström, Ä., Cöster, R., Höök, K., & Svensson, M. (2003, December). Paper prototyping a social mobile service. *MUM Workshop on Designing for Ubicomp in the Wild: Methods for Exploring the Design of Mobile and Ubiquitous Services,* Norrköping, Sweden.

Schön, D. A., & Bennett, J. (1996). Reflective conversation with materials. In T. Winograd (Ed.), *Bringing design to software* (pp. 171–189). New York: ACM.

Schrage, M. (1999). *Serious play: How the world's best companies simulate to innovate.* Boston: Harvard Business School Press.

Sellen, A., & Harper, R. H. R. (2001). *The myth of the paperless office.* Cambridge, MA: MIT Press.

Sengers, P., Boehner, K., David, S., & Kaye, J. J. (2005). Reflective design. *Proceedings of the CC 2005 Conference on Critical Computing.* New York: ACM.

Shami, N. S., Leshed, G., & Klein, D. (2005). Context of use evaluation of peripheral displays. *Proceedings of the INTERACT 2005 International Conference on Human-Computer Interaction.* Berlin: Springer.

Smith, I. (2005). Social-mobile applications. *IEEE Computer, 38*(4), 84–85.

Snyder, C. (2003). *Paper prototyping: The fast and easy way to design and refine user interfaces.* San Francisco: Morgan Kaufmann.

Szekely, P. A. (1996). Retrospective and challenges for model-based interface development. *Proceedings of the Third International Eurographics Workshop.* Namur, Belgium: Springer.

Trevor, J., Hilbert, D. M., & Schilit, B. N. (2002). Issues in personalizing shared ubiquitous devices. *Proceedings of the Ubicomp 2002 International Conference on Ubiquitous Computing.* Berlin: Springer.

Ulrich, K. T., & Eppinger, S. D. (2000). *Product design and development* (2nd ed.). Burr Ridge, IL: Irwin McGraw-Hill.

van Dantzich, M., Robbins, D., Horvitz, E., & Czerwinski, M. (2002). Scope: Providing awareness of multiple notifications at a glance. *Proceedings of the AVI 2002 International Working Conference on Advanced Visual Interfaces.* New York: ACM.

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

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009

Weiser, M. (1991). The computer for the 21st century. *Scientific American, 265*(3), 94–104.

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

Wisneski, C., Ishii, H., Dahley, A., Gorbet, M., Brave, S., Ullmer, B., Yarin, P. (1998). Ambient displays: Turning architectural space into an interface between people and digital information. *Proceedings of the CoBuild '98 International Workshop on Cooperative Buildings.* Berlin: Springer.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (2nd ed.). San Francisco: Morgan Kaufmann.

Woodruff, A., & Aoki, P. M. (2004). Push-to-talk social talk. *Proceedings of the CSCW 2004 Conference on Computer Supported Cooperative Work.* New York: ACM.

Yeh, R. B., Liao, C., Klemmer, S. R., Guimbretière, F., Lee, B., Kakaradov, B., et al. (2006). ButterflyNet: A mobile capture and access system for field biology. *Proceedings of the CHI 2006 Conference on Human Factors in Computing Systems.* New York: ACM.

Downloaded By: [Carnegie Mellon University] At: 01:55 24 March 2009