Techniques and tools for field-based early-stage study and iteration of Ubicomp applications: A dissertation proposal

Scott Carter
EECS Department
University of California at
Berkeley
Berkeley, CA 94720-1770
sacarter@cs.berkeley.edu

August 2, 2005

Abstract

Ubicomp applications are best evaluated in field settings that capture the range of contexts and interactions that they support. But Ubicomp applications are difficult to deploy, and evaluate. We propose to contribute innovative methods and rapid prototyping tools that overcome challenges with Ubicomp evaluation in field settings. In particular, we present tools and techniques that leverage pre-existing infrastructures and the increasing power of mobile devices to improve field evaluations of Ubicomp applications.

1 Introduction

Ubicomp applications, such as those that adjust output based on complicated contextual cues or involve public or mobile displays, are tightly coupled to the situations of their use. Because of this Ubicomp tools broaden our ability to enhance activities, but this makes the tools difficult to build and evaluate. In particular, Ubicomp applications are best evaluated in field settings that capture the range of contexts and interactions that they support. But building an application robust enough for field deployment can be problematic, especially when no infrastructure exists to support it. Also, evaluating people's use of applications in field settings can be time intensive.

In our review of Ubicomp application development, we identified challenges that Ubicomp application developers face when attempting to iterate and evaluate tools: scale, ambiguity, data sparsity, unobtrusiveness, and rapid prototyping [16]. Scaling deployments can be difficult because novel tools often need to be maintained constantly, and scaling evaluations can be difficult because researchers must organize and observe all study participants. Furthermore, many interactions of interest in a study are spontaneous or non-task-based and thus difficult to sense, leading to a sparsity of data when compared to the effort that went into building the tool. And though the goal of Ubicomp research is to seamlessly integrate technology into other activities, it is difficult both to build a tool that remains unobtrusive and to monitor the tool's use without effects on study data. Finally, few tools exist to aid the rapid iteration of Ubicomp tools and to resolve ambiguous input.

We propose to contribute innovative methods and rapid prototyping tools that overcome the seminal challenges with Ubicomp evaluation in field settings. In particular, we propose tools that reduce the need for writing software for testing Ubicomp applications in the field by using Wizard-of-Oz (WoZ) methods, make deployments more scalable by using pre-existing infrastructure and easily deployable devices and more unobtrusive by taking advantage of devices people already own and use, and facilitate organization and visualization of evaluation data. We also propose modifications and insights that extend the usefulness of scalable self-report methods for evaluation. With these approaches we can support the iteration and evaluation of a range of common themes in Ubicomp applications, including mobility, context awareness, and capture and access.

Our thesis is that **tools and techniques that leverage pre-existing infrastructures and the increasing power of mobile devices can improve field evaluations of Ubicomp applications. Specifically, we can address difficulties of scale by minimizing infrastructure deployment, lowering per-participant researcher-hours, and encouraging uptake; data sparsity by capturing and encouraging feedback from critical events; unobtrusiveness by using devices and interfaces with which participants are already comfortable and lowering participant distraction costs; ambiguity by allowing "wizards" to resolve ambiguities; and rapid prototyping by simplifying interface iteration, reducing coding, and reducing the burden of re-installations.**

In this proposal, we review other researcher's evaluations of Ubicomp applications and describe case studies in Ubicomp deployments. We then derive from those studies challenges for Ubicomp evaluations and discuss potential solutions. We then describe our approaches to these solutions, which concentrate on early stage, field-based evaluation, and our plans to use longitudinal studies and demonstration applications to validate these approaches. We conclude with a survey of related work and a timeline.

2 Ubicomp evaluation related work and case studies

To understand the important challenges in Ubicomp evaluation, we investigate prior work in two ways. First, we broadly review other researcher's experiences with evaluations at different stages. Second, we more closely inspect three evaluations that we conducted.

2.1 Related work

Ubicomp evaluation has been a major topic for discussion over the past few years, as evidenced by several recent workshops and special interest group discussions on evaluation of Ubicomp systems and sub-areas of Ubicomp such as peripheral displays [79, 81, 9]. Additionally, recent articles have remarked on the difficulties and challenges facing Ubicomp Evaluation. Holmquist *et al.* identified issues facing mobile Ubicomp evaluation, including bandwidth fluctuations, location-specific needs, ad-hoc interactions and users' adoption of different social roles for different situations [42]. Abowd *et al.* and Scholtz and Consolvo emphasize that new metrics need to be applied to evaluate Ubicomp applications given the loosely-structured nature of activities that they may support [5, 80].

As a result, a variety of interesting work is being done in Ubicomp evaluation. Despite this, the number of systems that have gone through true iterative design cycles is quite small. We have been able to find documented examples of only a small number of systems that included multiple design iterations (*e.g.*, [1, 67, 68, 91]). We begin by discussing these examples of iterative design, followed by a discussion of some places that existing formative and summative techniques have been applied, and a discussion of recent work in evolving and developing new evaluation techniques geared specifically to the domain of ubiquitous computing. We end with a discussion of why Ubicomp evaluation is difficult.

2.1.1 Examples of systems where multiple evaluations occurred

Examples of almost every major category of Ubicomp system from Abowd and Mynatt [4] have undergone iterative design, demonstrating the value of repeated evaluation, prototyping, and extended use across the domain of ubiquitous computing. One of the first Ubicomp systems to receive extensive study from an end-user perspective was eClass (formerly Classroom 2000) [1, 3]. EClass was a *capture and access* application that allowed students revisit the information presented during lectures. Lectures were given in a sensor-rich environment in which, at various times, lecture slides, written comments, video and audio, and other classroom activities were captured and organized. Students could access this information later via the World Wide Web (WWW). EClass was used and evolved over the course of more than three years. During this period, numerous qualitative and quantitative evaluations were completed, and the system itself evolved and changed. As a result of this work, and other related projects, Abowd *et al.* developed the concept of a *living laboratory*, an environment occupied by real users in which a research system is deployed, evaluated, and updated as it is used over time [2]. This was perhaps the first successful long-term deployment of a ubiquitous computing application, and certainly the first such deployment to include regular study of and feedback from end users. While eClass was wonderful proof of the potential of Ubicomp applications, and a great example of iterative design, neither the evaluation techniques nor the prototyping tools used in the project supported rapid iteration. Another early system, Tivoli [67], was a *natural interface* developed to support meeting activities. Moran *et al.*'s use experiences with Tivoli led them to develop easily tailorable tools so that they could better adapt the application to varying user needs.

The applications just described provide published examples of iterative design in Ubicomp, and all involved mainly summative evaluations. In contrast, both Jiang *et al.* and Mynatt *et al.* went through multiple iterations in the design of prototypes of Ubicomp applications. Jiang *et al.* developed and compared two prototypes of large-screen displays to support firefighters in *context-aware* incident command centers, and then developed a third, improved display based on the results [91]. Their evaluation involved showing the prototypes to firefighters and asking for feedback. Mynatt *et al.* developed multiple designs of a *context-aware* digital family portrait, all before a working prototype was completed [68]. They were interested in determining exactly what users might want in a display of activity levels for the family members of older people living alone at home. Their iterations involved a combination of techniques such as surveys, interviews, and WoZ evaluation [68].

In addition to these examples of iterative design, numerous developers have successfully used either formative or summative evaluation techniques in the design of Ubicomp applications. Below, we give examples of both. We present formative techniques separately from summative techniques because they have significantly different goals: understanding requirements (formative) versus evaluating working systems (summative).

2.1.2 Formative evaluation techniques explored

Formative techniques used for requirements gathering and understanding problem spaces, at the stage before actual systems are built, are probably among the most common found in Ubicomp application development. In the domestic arena, Crabtree *et al.* conducted ethnographies to determine appropriate sites in which to situate Ubicomp technologies, while Beckmann *et al.* and Venkatesh studied issues with the adoption and installation of current technologies to generate suggestions for future designs and guidelines [28, 87, 11]. Bellotti's and Smith's fieldwork studying mobile document and information workers inspired software designs embedded in current practice [13]. Bardram explored how Ubicomp tools may better integrate data access with patient care in hospitals [7] and Hayes *et al.* conducted interviews and participant observations to inform the development of wearable tools to aid those involved in aiding children with autism [40]. Also, researchers increasingly are using ESM and diary study methods for requirements gathering. Consolvo has conducted preliminary formative evaluations of Ubicomp technologies using ESM [24]. Carter has investigated the use of different media in diary studies [19]. Intille has developed tools to aid event-contingent ESM [47].

It is less common to see formative techniques applied to the very early stages of prototyping and design, after requirements gathering is completed. WoZ prototyping is perhaps the most commonly applied technique. An early piece of work in early-stage evaluation of a non-working prototype was Oviatt's use of WoZ evaluation to test a multi-modal map application which combined speech and pen input in different ways [71]. Oviatt showed that problems with speech-only map interactions were assuaged when using multimodal input. Dählback was first to highlight the uses of WoZ systems in this domain in his description of pros and cons of WoZ, and his experiences applying it to an intelligent user interface [30]. Although Dählback's argument was in the context of intelligent user interfaces, the major pros (low cost, ability to gather empirical use data at an early design stage) and cons (ethical problems regarding deceiving users, potentially artificial use cases) he discussed are also relevant to Ubiquitous computing (which often uses intelligence). Additionally, WoZ evaluation has been applied to the design of ambient displays and other Ubicomp systems [26, 68]. Benford *et al.* showed that participant reports are useful in the simulation of location information for mobile applications [15]. Finally, Jiang *et al.*'s firefighter system (described above) was evaluating using a form of user-based expert review modified to evaluate non-working systems [91]. Although his work did not explicitly explore the pros and cons of this technique, it does illustrate the technique's applicability.

2.1.3 Summative Evaluations

In contrast to formative evaluations, more varieties of summative evaluations have been applied to Ubicomp systems. These can be broken into laboratory studies, and evaluations done in unconstrained settings. Laboratory studies can feature controlled variables and statistically significant data. They have been used most successfully to explore specific questions such as the relative value of different input/output methods (an example described below is McGee *et al.*'s study of post-it notes vs. a multi-modal system [65]). Field studies or studies of unconstrained use of a system, in the lab or the field, may provide more realistic information about the problems and advantages that users will enjoy. While this trade-off exists even with desktop applications, it should be noted that Ubicomp systems are particularly hard to test in the lab because their use is often highly contextualized – they must often function in multi-person, multi-task settings.

Controlled studies McGee *et al.* compared existing military command post paper tools (maps and Post-it notes) with a tangible multi-modal system called Rasa [65]. In a controlled, laboratory study, participants were asked to use Rasa in a 90-minute simulation that included a simulated power failure. Results showed that users preferred the multi-modal system, because it enhanced their existing task with computing but did not detract from the flexibility of their existing paper tools.

A sub-area of ubiquitous computing, peripheral displays, has received much evaluation attention recently. Peripheral Displays refer to applications that are *not* meant to be the focus of the user's attention. Instead, they either notify a user about important events (alerting displays), or allow a user to monitor information (ambient displays), while the user is attending to a primary task or activity. These types of displays are often combined, supporting continuous monitoring mixed with occasional alerts.

Mamykina *et al.* conducted a summative evaluation of ambient displays that enhance a user's ability to control their pace during time-sensitive, cognitively demanding activities [57]. The researchers evaluated two activities in a lab setting: taking a test and giving an oral presentation. They tested several display designs, all indicating how much time the participant was spending on each question or slide relative to the time allotted for the test or presentation. The study demonstrated that ambient technology can successfully control pacing in cognitively demanding activities. This study is unique among ambient display studies because of its strong results in a controlled, laboratory setting.

Researchers investigating alerting displays have conducted more evaluations, many focused on understanding the impact on user attention of different types of alerts or animations. Cutrell *et al.* investigated instant messaging interruptions at various times during different activities, finding them to be most distracting in the beginning of a new task and especially for fast, stimulus-driven search tasks [29]. Intille explored change blindness as a new model for displaying information, in order to minimize detectable motion and preserve a calm environment [46]. Maglio explored the design of scrolling displays, finding that scrolling is more distracting to a high-attention primary task than displays that start and stop, but information in both is remembered equally well [56]. McCrickard *et al.* found that animations like fading, rolling, and tickering were not distracting to low-attention primary tasks [63]. Bartram *et al.* also studied detection and distraction of motion in interfaces [8].

McCrickard *et al.* explored a more general framework, the IRC framework, for evaluating alerting displays, applying three metrics they identify for describing user notification goals — interruption, reaction, and comprehension — to a lab evaluation of the Scope interface [64]. They found that questionnaires based on the IRC framework suggested redesigns similar to those carried out earlier by the Scope design team, which were based on other questionnaires as well as more time-intensive user studies and expert reviews [55].

Alerting display research has also investigated interruptibility. For example, in a WoZ study, Hudson *et al.* explored how robust sensor-based predictions of interruptibility might be constructed, and which sensors would be most useful to such predictions [43]. They found that relatively simple sensors can probably achieve 75-80% accuracy for estimating human interruptibility.

Unconstrained studies Unconstrained studies are those that value realism over controlled settings and can be used to gain qualitative feedback about nuanced or context-specific use of an application. Oviatt used a wearable computer to test the robustness of speech in unconstrained settings [72]. When combined with pen input, the system was remarkably effective in recognizing speech. Consolvo *et al.* used Lag Sequential Analysis in a summative evaluation to gather quantitative data about how their system was used [25]. Although this technique had not been applied to Ubicomp before, it proved interesting and informative. One major advantage of the technique is that it does not require researchers to be physically present with users, while a disadvantage is its requirement that hours of video be hand coded.

Beckwith interviewed and observed inhabitants of a nursing home to gather qualitative data regarding the inhabitants' perceptions of sensing technologies installed in the home [12]. Crabtree conducted an evaluation of a mixed reality mobile phone game via analysis of logged data combined with extensive participant observation [27]. The application of these standard qualitative methods is useful in uncovering social activities and perceptions surrounding Ubicomp technologies.

Peripheral displays, particularly ambient displays, have been studied in unconstrained settings. Unfortunately, details are rarely reported, or results focus on informal feedback about design innovation. For example, Change *et al.* report only qualitative feedback from an undisclosed number of short-term users who were not emotionally involved with each other in their evaluation of the LumiTouch system, which includes a pair of picture frames augmented with touch sensors and LEDs through which remote loved ones can communicate [23]. Mynatt *et al.* evaluated Audio Aura, which provides information about e-mail and colleagues' activity via background auditory cues, by introducing the system to nine volunteers who provided only their initial, qualitative impressions of the system [69].

Equally as informal, some ambient displays have been exhibited in museum or gallery settings where they were used by hundreds of users but never tracked in detail. One such project, exhibited at the San Jose Museum of Art, is the Oxygen Flute, a chamber in which a person's breathing is sensed and transformed into music [21]. Another exhibit shown in the NTT-ICC museum in Tokyo was Pinwheels, which visualizes various streams of information such as people's movements in physical space (*e.g.*, car traffic or movement in elevators) and in digital space (*e.g.*, e-mail exchanges or stock market transactions) [48]. Finally, the interactive poetic garden installation at the MIT Media Lab projects flowing words onto a pond surface and enables viewers to control the movement of words via a touch-sensitive pad [89]. The most common "finding" of exhibits and empirical studies like these is that users are interested in and excited by innovations in ambient displays.

Other unconstrained studies include the iterative design work of Abowd *et al.* and Moran *et al.* discussed earlier [1, 67]. Additionally, many of the published Ubicomp systems found in the literature include some feedback from users, either the developers and their colleagues, or less affiliated users.

2.2 Case studies

The previous section presented a high-level review of past evaluations of Ubicomp applications. In this section we present three in-depth case studies of evaluations of Ubicomp systems [18]. The first, PALplates, was developed in 1996. The second, a nutritional tracking system, was designed and evaluated between 2001 and 2004. The third, Hebb, our keystone study, was developed and evaluated from 2002 through 2004. Our goal in presenting these evaluations is to show by example some of the difficulties that stand in the way of Ubicomp evaluation.

2.2.1 PALplates

Mankoff's and Schilit's PALplates [61] was intended to support office workers in doing everyday tasks by presenting key information and services at locations where they were most likely to be needed. The goal of the project was to create a Ubicomp system that could provide location-based services without requiring users to carry around a mobile computing platform. For example, suppose a user goes to the printer room to pick up a print job, and notices that the printer is almost out of paper. She might use the print room PALplate to request extra paper or resubmit her print job to a different printer. In contrast, the meeting room

PALplate might be used to make reservations for that room for a follow up meeting to one just completed.

PALplates were interactive computing platforms, deployed around the work place at *places of need*. Each PALplate provided unique services, based on its location in the office place. The goal was to create a system that, although only available at certain places, would feel ubiquitous to a user. The researchers hoped to do this by placing a system at each place the user was likely to be doing a task that might require computational support. This could free a user from the need to constantly carry a device with herself.

Because this was a novel approach to ubiquitous computing, the researchers wanted to get feedback from users as soon as possible about the effectiveness of such a system. They wanted to know if people would use the devices, and whether the way that devices were used were location specific. They also wanted to know whether users wanted additional services and whether they found the services in our initial set useful. Their goal was to generate requirements for our next iteration.

It was best to let users experience this novel approach to providing "ubiquitous" computational support. Yet implementing even a prototype of the system ubiquitously would have required installing infrastructure in many locations. Instead, the researchers decided to evaluate a paper prototype of the system in the field.

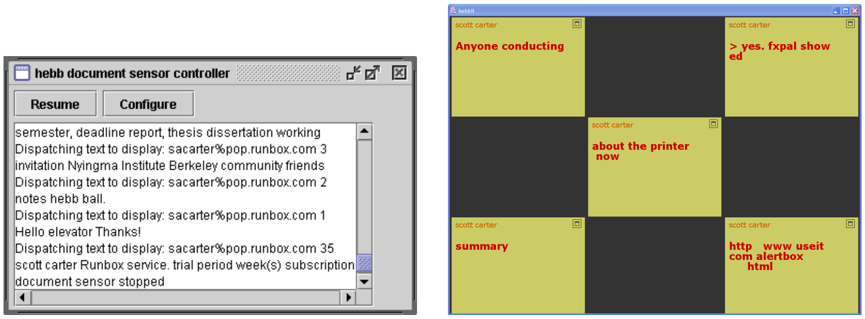
Overall, paper prototyping worked in this case. It allowed the researchers to explore whether providing computing services at a point of need was a viable idea, and it helped test four services and generate new requirements for additional services such a system would need to support.

2.2.2 Nutrition Tracking

Mankoff *et al.* built an application that uses inexpensive, low-impact sensing to collect data about what household members are purchasing and consuming, and used simple yet persuasive techniques to suggest potential changes [59]. Healthy eating can help to reduce obesity, and consequently the chance of developing chronic diseases such as diabetes [39]. Yet many people do not know exactly how many servings of fruits, grains, vegetables, and fats they are eating, or which nutrients are missing in their diet.

The proposed system gathered data about purchasing habits when receipts were scanned in with a handheld scanner (*e.g.*, when bills are being sorted at the end of the week). A shopping list, printed at the user's request, provided annotated suggestions for slight changes in purchases. This portable piece of paper provided suggestions for a more balanced diet (based on USDA guidelines) at the most pertinent moment: when the user was making purchasing decisions. It could encourage healthier purchases such as baked tortillas instead of chips, or wheat bread instead of white. The system also displayed a food pyramid skewed to indicate the relative amounts of different foods a user was purchasing, enabling him to easily identify areas needing change.

As with PALplates, the goal was to provide users with a computational service without requiring them to constantly carry a computer with them. Again, the researchers wanted to know if people would use the device: was the printed shopping list sufficient to meet a person's



```
semester, deadline report, thesis dissertation working
Dispatching text to display: sacarter%pop.runbox.com 3
invitation Nyingma Institute Berkeley community friends
Dispatching text to display: sacarter%pop.runbox.com 2
notes hebb ball.
Dispatching text to display: sacarter%pop.runbox.com 1
Hello elevator Thanks!
Dispatching text to display: sacarter%pop.runbox.com 35
scott carter Runbox service. trial period week(s) subscription
document sensor stopped
```

scott carter
Anyone conducting

scott carter
> yes. fxpal show
ed

scott carter
about the printer
 now

scott carter
summary

scott carter
http   www useit
com alertbox
html

(a) (b)

Figure 1: Hebb **(a)** interest sensor and **(b)** public display. The interest sensor scans participant's e-mail for topics of interest. The public display collects topics and displays those likely to encourage communication and collaboration between groups.

needs while shopping, and were recommendations useful? The researchers also questioned whether the process of using our device fit well into a person's everyday shopping patterns. Lastly, they questioned whether people could easily interpret the information displayed by the food pyramid. Answering these questions was challenging. Because the system needed to integrate into a person's shopping pattern, they had to deploy a working prototype in a field setting, a home. Additionally, the recommendation system required historical information and used a complex algorithm. This could not easily be simulated. Despite this, they wanted to test out as many aspects of the system as possible ahead of time, before putting too much effort into prototyping.

In retrospect, the researchers had to spend a lot of time and effort on this system before learning which aspects of it were truly useful, and which were not. The initial interviews conducted had limited value when compared to what what was learned when the researchers actually deployed the system. The paper prototyping was very successful, but only allowed them to answer a very narrowly defined question. The deployments were highly informative, despite being crippled by problems with placing a very early prototype in the field. However, the cost of doing them was unduly high. Simulation appears to be a viable alternative that should be considered.

2.2.3 Hebb

It is not uncommon for multiple small working groups in a large organization to overlap in the problems they are trying to solve, interests, or hobbies. Yet often such shared interests go undiscovered even among loosely coupled groups that share common spaces or are located in the same building. Better awareness could lead to fruitful collaborations, friendships, and other positive social change. Our approach was Hebb, a system designed to capture and convey shared interests [20]. This system arose from a series of formative interviews that we conducted with members of six small working groups. From these interviews, we found that the benefits of collocation (awareness of shared interests) often do not extend beyond the group. In response to this issue, we designed a system that senses group member's interests via e-mail analysis software and displays relationships between members on public and private displays to encourage conversation about those topics.

The Hebb system includes interest sensors, presence sensors and public and private displays (see Figure 1). The interest sensor generates name and keyword event pairs. Keywords are generated from user e-mail. The interest sensor also makes available encrypted full document data for use on personal PDAs. The presence sensor generates unique user identifiers (UIDs) based on users sensed in the space via either RFID badging or presence of the user's PDA on the local wireless network. The public display generates events indicating from which document keywords were recently displayed. Servers on each component and a remote discovery server handle subscription and networking.

We built Hebb because we wanted to know if it was possible to help loosely-coupled small groups build a better awareness of cross-group shared interests. Although we could test portions of the system in the lab, the only way to know if it really changed group dynamics was to deploy it in the field, over a long period of time. Additionally, the complexity of the system and the interaction of multiple devices distributed across both time and space and within the context of other activities made it next to impossible to design a lab study that would give us useful feedback.

Overall, we found it difficult to deploy this system both because participants had difficulty understanding the model of interaction and because it was difficult and time consuming to discover errors and update the system in the field. However, we found that users began adapting to the system rapidly and derived value from it.

3 Challenges for Ubicomp evaluation

Many of the lessons learned from reviewing related work and evaluating the three case studies overlap. In particular, issues of ambiguity, scale and data sparsity, integration into daily life, unobtrusiveness, and prototyping all were evident in several different evaluations. While some of these lessons may apply generally to the evaluation of any application, our evaluations have shown that they are particularly exacerbated in Ubicomp applications [16].

It is important that these issues be addressed to expand the scope of future Ubicomp evaluations. In their review of five different usability experiments, Gray and Salzman highlighted the importance of using larger sample sizes [36]. The authors argued that using larger sample sizes better supports statistical conclusions, appropriate participant randomization and settings conditions for internal validity and replication of studies for external validity. In order to expand the scope of Ubicomp studies, it is important to determine how to address the challenges we outline in this section, including how to measure appropriate metrics, scale deployments to more sites, increase the amount of data collected per site, and integrate the application into situations such that data can be collected unobtrusively.

Challenge # 1 – Rapid prototyping One reason for lack of iteration is the difficulty of quickly building working Ubicomp systems needed by most applicable existing evaluation techniques. Abowd and Mynatt [4] discuss several reasons for this, including that effective evaluation, in which users are observed interacting with the system in routine ways, requires a realistic deployment into the environment of expected use (a point also emphasized by Davies and Gellersen [31] in their review of the lessons learned from successful deployments). Developers felt this was problematic: "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." Development is further complicated because applications often depend on extensive infrastructure, sophisticated sensing, or the existence of other systems. Thus, developers currently must put significant effort into an application before testing it, leading to less iteration. Better rapid prototyping tools in support of early-stage evaluation techniques such as those we are proposing to build could help to address this issue.

Challenge #2 – Scale Ubicomp systems typically must handle issues of scale not faced by desktop systems, functioning across multiple devices, locations, or over long periods of time or across multiple users. As mentioned, eClass included multiple projected displays for the instructor, a large-screen, rear-projection whiteboard, pen tablets for students, video and audio recordings, and web based access to recorded data at a later time [1, 3]. Ubicomp applications could in principle be used by the number of users visiting a popular website. How does one evaluate systems for thousands of users before they are in use by that many? How about after?

Looking back at our lessons learned, issues of scale arose over and over again. For example, our paper prototypes had trouble scaling across time and amount of data. Other early stage techniques such as heuristic evaluation [70] would face similar problems in scaling across the number of devices and scenarios for which Ubicomp systems may be designed. In deployment, we found it difficult to scale the number of devices and users due to the many challenges of maintaining and supporting our system. We learned too at the deployment phase that a local informant can help speed up acceptance and help mitigate time spent on system maintenance, reducing the amount of time we needed to spend on each installation and facilitating scaling the system.

Challenge #3 – Ambiguity Ubicomp systems are typically sensing-based systems. As Bellotti *et al.* discuss in their article, "Making Sense of Sensing Systems" [14], this can lead to serious usability problems. Research remains to be done in identifying the best user-interface mechanisms for dealing with sensing issues such as ambiguity and errors [32, 41]. However, this work is best done in conjunction with evaluation techniques and tools capable of helping to judge it.

Our case studies above show that ambiguity and errors are serious and important issues in determining system success and understanding usability problems. When Ubicomp systems depend on recognition, recommendations, machine learning, or other ambiguity-prone components, evaluation techniques need to provide feedback on acceptable levels of accuracy. Feedback is also necessary to understand error recovery. During deployment, help is needed in mitigating inevitable errors and misunderstandings. If ambiguity has not been honed to an acceptable level, users are likely to quickly decide a system is not worth using.

Challenge #4 – Data sparsity It is difficult to gather data in Ubicomp evaluations for a number of reasons: evaluations are difficult to scale, events of ambiguous importance may go undetected, and critical events are often spontaneous and sparse. According to Holmquist et al. (2002), Ubicomp evaluation is difficult because "small bursts of usage are often extended throughout the entire day, and in many different places." Developers that we interviewed also cited frustration with this issue. For example, in the case of mobile applications, developers mentioned that use was irregular and fleeting, leading to small amounts of data even after they put in the effort to make a field deployment possible. To aid data gathering, tools and techniques are needed that enable large-scale deployments that increase the total quantity of data gathered by including more users or running for longer periods of time; can capture and encourage feedback about critical events; and can aggregate data about critical events into a coherent picture of application use.

Petersen *et al.* encountered this challenge in their work studying the evolution of television use by families in their homes over the course of six months [73]. Because of time and privacy concerns it was difficult for the researchers to obtain first-hand data of the events that led to changes in the way families used the studied technology. To better understand these events, researchers experimented with several methods that would help reconstruct the user's experiences, including "scenario-framed hands-on use sessions," role playing, and diary studies. The researchers found that the use sessions and role playing were helpful but that participants found the diary study to be too time consuming.

Challenge #5 – Unobtrusiveness We found in our latter two case studies that evaluation techniques failed to appropriately handle integration into daily lives. This challenge arose in two ways. First, it is often difficult to obtain the feedback necessary to evaluate an application while maintaining that application's unobtrusiveness. We encountered this difficulty while evaluating the Hebb peripheral display. As Weiser states in his seminal article on Ubiquitous Computing: "Most of the computers will be invisible in fact as well as metaphor" [88]. A common type of "mostly invisible" display, peripheral displays are designed to show information without distracting. However, it is difficult to design an evaluation of the display that does not call attention to the display itself, thereby breaking its peripherality.

The second piece of this challenge is that an evaluation of a prototype that has not yet evolved to be unobtrusive will yield significantly different results from one that has. This point is informed by the idea of invisibility, or readiness-to-hand [90]. As Tolmie [85] and Star [78] have discussed, a technology becomes invisible in use when it is no longer remarked upon as novel nor breaks down. Put another way, a technology has achieved invisibility when it becomes ready-to-hand (*e.g.*, a technology perceived as an extension of the body, such as working mouse) rather than present-at-hand (*e.g.*, a mouse that sticks often). A technology that is unfamiliar, that constantly suffers breakdowns or re-installations, is unlikely to become ready-to-hand.

In our studies of Hebb, we found that unobtrusiveness was exacerbated by interruptions caused by reinstallations. In addition, in our studies of the nutrition project showed that the degree unobtrusiveness of an application is only apparent when the application is used and evolved in the appropriate context. In particular, we found it difficult to predict how a system would integrate into daily life from our evaluation of the paper prototype. This problem was not surprising given the evaluation technique we chose. We simply note that it is more serious because Ubicomp systems are typically integrated into everyday life or other, external tasks, than with single user, task oriented desktop systems. This challenge has been taken up in the design of collaborative systems. Work by Steves *et al.* and Pinelle *et al.* found that early stage evaluations conducted out-of-context can aid the usability and adoption rate of later field studies [84, 74]. Still, it is difficult to foresee all of the issues and conflicts a technology will raise until it is introduced to a field setting.

4 From challenges to solutions

The challenges for Ubicomp evaluation, rapid prototyping, scale, data sparsity, ambiguity, and unobtrusiveness, require numerous innovations to solve. Work needs to be done in all stages of evaluation, including formative, early-stage, and summative, as well as in various methodologies, including lab studies, simulations, and field experiments. In this section, we describe our focus on **formative and early stage field-based** methodological approaches to evaluations of applications that involve **asynchronous, distributed** activities, as well as our concentration on **mobile devices**. We then describe solutions to challenges with these evaluations and explore the approaches taken to address these challenges in past work.

4.1 Application coverage

Ubiquitous computing spans a wide variety of applications. Since no work has proposed a theoretically grounded taxonomy of Ubicomp applications, we rely on distinctions of application support used in the closely-related Computer-Supported Cooperative Work literature: distributed versus collocated and synchronous versus asynchronous [6]. We also characterize our contributions based on types of interfaces (public, peripheral, mobile, and tactile) and sensors (implicit and explicit).

We cannot hope to support evaluation and rapid iteration for all types of Ubicomp applications. Instead, we focus our support on those applications that support asynchronous, distributed activities. Furthermore, we also contribute to mobile display interface development, provide support for explicit sensor input (*e.g.*, data provided directly by participants), and take advantage of past work to support implicit sensor input (*e.g.*, data collected seamlessly by other technologies).

Different applications require different types of evaluations, which can more or less difficult. Some Ubicomp applications lend themselves to evaluation more than do others. In particular, those that support primarily synchronous, collocated activities, such as living laboratories and classroom augmentations, are more straightforward to evaluate [2]. Collocated and synchronous activities are easier to observe, simplifying data collection. Ease of observation and data logging also make evaluations of these applications less intrusive. Still, these applications are difficult to prototype and scale. Ubicomp applications that support primarily asynchronous, distributed activities present more difficulties for evaluation. In addition to challenges prototyping and scaling these applications, data sparsity and unobtrusiveness are particularly difficult [27]. With these applications, users tend to be mobile and to change context often, making data collection more difficult. Also, these applications tend to require different views on data that match the different contexts in which a user might view them, making rapid interface iteration more difficult.

Our application iteration support concentrates on mobile displays. Other work supports the creation of peripheral [62] and tactile [37] displays. Furthermore, many interactive public displays could be prototyped rapidly using a wide variety of standard applications, such as web browsers or browser plug-ins (*e.g.*, Flash). However, our tools could support *evaluation* of public and peripheral displays (in particular, participant reports of use).

Finally, our input support concentrates on self reports from participants, but other sensors can be integrated into studies using the Context Toolkit. Also, we do not provide any explicit support for applications the require tangible input [51] – our experience is that the asynchronous, distributed activities we seek to support rarely require it.

4.2 Methodological approach

Our work concentrates on formative and early stage field-based methodologies to approach challenges for asynchronous, distributed Ubicomp applications. Formative and early stage methods facilitate informing designs with user feedback at precisely the point when feedback is most useful. In later stages, while user feedback is crucial at indicating whether a design was or was not successful, it is more difficult to change a system's design. Also, researchers investigating the viability or social impact of potential applications may never need to create completely functional systems: early prototypes may provide enough insight, for example, to develop guidelines for designers. We further narrow our contribution to field-based methodologies, because asynchronous, distributed Ubicomp applications are situated and context-dependent, experimental realism is crucial for evaluation. While other researchers have investigated recreating field situations in lab settings [50], field experiments are widely accepted as the methodology that provides the most experimental realism [66].

Challenge #1 – Rapid prototyping To facilitate rapid prototyping for Ubicomp field evaluations, a solution should **simplify interface iteration**, **reduce coding**, and **reduce the burden of reinstallations**. The first two solutions are important in any prototyping system. Reinstallations are particularly important to support in Ubicomp applications being field tested.

Challenge #2 – Scale: To scale a Ubicomp field evaluation, a solution should **minimize infrastructure deployment**, **lower per-participant researcher-hours**, and **encourage uptake**. Most Ubicomp systems require that complex infrastructures be in place: a solution should either make infrastructures easy to build or should make use of already-available infrastructures. When scaling an evaluation, the time needed for researchers to conduct the study tends to scale as well. A solution should use consent procedures and evaluation methods that minimize researcher burden. Finally, to encourage uptake, a solution should facilitate building applications that have benefit to users [38].

Challenge #3 – Data sparsity: To aid data gathering in a Ubicomp field evaluation, a solution should **capture and encourage feedback from critical events** or **encourage increasing the scale of a deployment**. In many Ubicomp deployments, critical events are spontaneous and sparse, so it is important to capture and gather feedback on as many as possible.

Challenge #4 – Ambiguity: To handle ambiguity in a Ubicomp field evaluation, a solution should **resolve ambiguities** or **indicate ambiguities to users**. Ambiguities could be resolved either programatically [60] or via a person "playing computer" (a wizard in WoZ studies). A wizard could also indicate ambiguities to users, or a solution could have functionality to store and expose ambiguous data.

Challenge #5 – Unobtrusiveness: For a Ubicomp field evaluation to be unobtrusive, a solution should **use devices and interfaces with which participants are already comfortable** and **lower participant distraction costs**. If new devices and interaction paradigms are introduced to an environment, users will need time to learn and understand the technology and may reject it outright. If an application can make use of devices with which participants are already comfortable, rapid uptake is more likely. For the same reasons, evaluations also should minimize distractions and interruptions.

4.3 Tailoring evaluation techniques to Ubicomp

Based on our survey of related work presented earlier, it appears that while many techniques exist for early stage requirements gathering, only a limited set of existing techniques are being applied to the early stages of prototyping and design, after requirements gathering is completed. The main early-stage prototyping technique that has been applied successfully is WoZ evaluation [72, 68]. At the same time, a variety of techniques are being applied successfully at the summative stage. This imbalance suggests that there may be a dearth of lightweight techniques suitable for evaluating early prototypes of Ubicomp systems. A lack of lightweight techniques may inhibit iteration because designers must put significant effort into a prototype before they are able to test it or find out about problems with it. Two possible solutions present themselves: One is to create or modify lightweight, early stage evaluation techniques so they are suitable for testing out early prototypes of Ubicomp systems. The other is to create rapid prototyping tools that make it easier to quickly build prototypes suitable for more complex evaluations.

An examination of the literature shows innovation in both early-stage, lightweight techniques such as paper prototyping and heuristic evaluation, and summative techniques such as lab studies. Additionally, some work on rapid prototyping tools is taking place. Below we examine each of these.

4.3.1 Innovating lightweight techniques

Chandler *et al.* developed a set of guidelines for paper prototyping of multi-modal applications [22]. This work was then expanded to support lightweight WoZ evaluation, using a technique called "Multimodal theater" [82]. Although not entirely the same as Ubicomp, multi-modal applications share similar issues with regards to the variety of input and output modes they typically must support. In a separate piece of work, paper prototyping was compared to an interactive system for evaluating the design of a kitchen support system [54]. They found that more people were needed to run the paper prototype study (making it less lightweight), and that it was hard to make sure that it was present and interactive at appropriate times. Carter and Mankoff also found that while paper prototyping allows for rapid iteration, it's lack of interaction support significantly limits what metrics it can test [18].

Researchers have created modified versions of Heuristic Evaluation for different domains: Mankoff *et al.* modified HE for peripheral displays and Po *et al.* modified HE for mobile applications [58, 75]. Other researchers have created or adapted different techniques for use in formative Ubicomp evaluation settings. However, these approaches do not concentrate on situated application use and thus are outside the scope of this thesis.

Other researchers have investigated reporting methods for field evaluations. Hutchinson *et al.* experimented with a new form of evaluation called Technology probes [44]. "Technology probes are simple, flexible, adaptable technologies with 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." Another lightweight technique is Intille *et al.*'s use of images as prompts for questions in event-contingent ESM studies [45, 77]. Except for setup costs, this technique has the advantage of aiding participant's memory while not requiring additional ongoing effort from end users or experimenters. The trade-off is that it depends upon infrastructure to capture images and send them to the user. Also, the sampling questions are not explicitly tied to issues with applications.

Another form of report methods, diary studies, remains relatively unexplored. The diary study is a method of understanding participant behavior and intent that reduces researcher burden by having participants record events as they happen. While others have used diary studies to understand participant behavior [76], because they tend to be burdensome for both participants and researchers, researchers rarely use them to evaluate Ubicomp applications. Also, no work has explored the impact of using different types of media, including new digital recording media, on the diary study method.

Overall, prior work in lightweight techniques suggests that there is still room for improvement. In particular, there is no support for iterating interfaces as rapidly as paper in an interactive medium. This is a particularly difficult problem when testing a mobile application in a field setting as use will be distributed. There is also more work to be done using reporting methods for field evaluations of asynchronous, distributed Ubicomp applications. In particular, experience-sampling could be made more scalable and be better linked to application development, and the diary study method could prove useful if made less intrusive.

4.3.2 Innovating later stage techniques

Trevor *et al.* developed a comparative study methodology similar to a laboratory experiment [86]. They used quantitative and qualitative data to compare and contrast interfaces that included different hardware and software and were deployed in different environments. The difficulties of evaluating Ubicomp applications made it impossible for them to conduct a true controlled, laboratory study. However, their interfaces were designed for evaluation, and this allowed them to gather information that could be used for comparison. Trevor *et al.* gathered data about standard usability issues such as usability and utility. 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. However, their evaluation concentrated on gathering feedback from complete systems rather than iterating at early stages of development.

The living laboratory is another later stage technique that seeks to test Ubicomp systems in an everyday context. EClass, included multiple projected displays for the instructor, a large-screen, rear-projection whiteboard, pen tablets for students, video and audio recordings, and web based access to recorded data at a later time [1, 3]. Beaudin *et al.* have developed a rapidly deployable sensor network with the goal of extending the living laboratory technique to different field environments [10]. However, this technique involves collocated activity, and is not the focus of our work.

Recent work suggests that recreating the context of use through scenarios in lab settings may provide as much feedback on usability problems as field experiments for some constrained Ubicomp applications. Kjeldskov *et al.* found that a laboratory test approximating field use yielded more found usability problems at a lower cost than field experiments [50]. In comparison, our work focuses on highly situated and distributed activities that are difficult or impossible to recreate in laboratory settings. Also, Po *et al.* found that in a Heuristic Evaluation evaluators judging an interface based on a use scenario found about as many context-specific usability problems as did evaluators judging the interface in the field [75]. Again, our work focuses on gaining feedback from users rather than *pre-hoc* evaluations.

4.3.3 Rapid Prototyping

Researchers have developed some tools and toolkits to allow developers to rapidly prototype Ubicomp devices for early-stage testing. Li *et al.* developed Topiary, which allows Ubicomp application designers to rapidly prototype location-enhanced applications [52]. Lin developed Damask, which uses the concept of design patters to allow designers to rapidly develop interfaces for multiple devices [53]. Sinha's and Landay's CrossWeaver allows designers to informally prototype multimodal, multidevice user interfaces [83]. Greenberg and Fitchett developed Phidgets to support development of tangible interfaces [37]. Klemmer *et al.* developed a toolkit to support prototyping tangible input (for example, from RFID tags or bar codes) [51]. Matthews *et al.* developed a toolkit that supports rapidly prototyping peripheral displays [62].

Of these tools, the two that have the most in common with those that we are proposing are CrossWeaver and Topiary. Using Crossweaver, researchers can use WoZ techniques to rapidly test multi-modal, multi-device applications in laboratory settings. However, Crossweaver supports neither scalable field deployments nor evaluations. Researchers can also use WoZ techniques in Topiary, in particular to update location information. Furthermore, Topiary is designed to support field evaluations. However, Topiary does not support large scale evaluations. Also, there are no reporting methods integrated into Topiary, nor is it designed to interact with other non-location-specific aspects of an application. Finally, while Topiary allows a researcher to change rapidly the behavior of a mobile application, it does not provide a visualization of the look-and-feel of the mobile interface to participants.

5 Field-based evaluation tools and techniques

It is critical to gather field data in both formative and summative stages of Ubicomp application design. In formative stages, field data can help translate user needs into new designs. In summative stages, field data can elucidate how designs operate in context and can aid with iterating designs. However, field data is not often collected for two key reasons: field studies are resource intensive to conduct and Ubicomp applications are difficult to build and deploy. To address these issues, we modify and extend self-report evaluation methods and support application iteration and deployment.

In this section we introduce two approaches designed to address challenges in Ubicomp evaluation. The first approach concentrates on the early stages of iterative design in which a developer builds a tool, evaluates its use in a field context, and redesigns and redeploys the tool. Again, there are many approaches to this problem. A common theme is to build a toolkit that lower the bar for developers to build new iterations [33] or sketch-based interactive versions of a tool [52]. In our approach, we take advantage of the large infrastructure of increasingly powerful mobile devices and messaging services to deploy rapidly WoZ based tools.

The second approach concentrates on the formative stages of application development, in which the goal is to understand the context in which an application will be used. There are several methods that could be used for this goal, including direct observation, interviews, diary studies, *etc.* We chose to focus on diary studies and experience-sampling studies because they are participant-driven and thus easy to scale. Our contributions include recommendations for the use of recording media in diary studies as well as technique and tools that lower the burden on participants while increasing the quality of data gathered for both diary and experience-sampling studies.

5.1 Wizard-of-Oz for rapid prototyping

Gathering field data for design iteration presumes the existence of an application to evaluate. But we and others have identified difficulties with both application construction and deployment. Kjeldskov and Graham note that there is an “unwillingness to implement mobile systems which are uncertain to succeed and take a long time to evaluate and implement” because of “the current cost of such technology and the associated implementation overhead” [49]. Furthermore, as noted above, when deploying Ubicomp applications it is difficult to maintain unobtrusiveness, manage reinstallations, and find infrastructure support [18, 31].

Figure 2: A wizard using Momento (right) while concentrating on another task.

Our solution, Momento, makes use of common infrastructure and devices for mobile applications [17]. Momento makes prototyping easier by **simplifying interface iteration**, **reducing coding** required to deploy an application, and by making use of established network platforms to **reduce the burden of reinstallations**. To facilitate scaling evaluations, it **minimizes infrastructure deployment** by making use of a pre-existing infrastructure, **lowers per-participant researcher-hours** by providing an interface to visualize data from large numbers of participants, and **encourages uptake** by simplifying participant sign-up. Momento addresses data sparsity by allowing researchers to **capture and encourage feedback from critical events** sensed via Bluetooth sensors. It allows researchers to **resolve ambiguities** using a human wizard. It facilitates unobtrusiveness by **using devices and interfaces with which participants are already comfortable**. Finally, it **lowers participant distraction costs** by using context data so that participants are not interrupted inappropriately.

5.1.1 Completed work

Interviews we conducted with three developers of mobile systems included many mentions of barriers to iterative design. In addition, they highlighted three key resulting difficulties for evaluating mobile systems in the field. The developers most often cited frustration with data *sparsity*. In particular, they mentioned that the highly situated nature of mobile applications meant that use was irregular and fleeting, mitigating their ability to gather enough data to iteratively develop prototypes. In effect, they felt even when they developed a solution robust enough for field use, the limited *scale of deployment* possible meant that they received only minimal feedback of actual use. To a lesser extent, developers also felt that the lack of familiarity with devices made those devices less *unobtrusive*. This reduced the validity of the resulting data, because participants were not habituated to the devices. Finally, they expressed the importance of *rapidly iterating* high fidelity interfaces based on participants' experiences with them in field experiments. Even once developers chose a particular platform for experimentation, rapid iteration was difficult because of a lack of rapid prototyping tools that could be used for field deployments of mobile applications.

We have created a tool, MObile Messaging and EvalutioN TOol (Momento), that addresses difficult issues in Ubicomp evaluation, including prototyping, scale, data sparsity, ambiguity, and unobtrusiveness [17]. Momento allows developers to make use of devices participants already own. Because of this, our tool can potentially increase the *scale* of studies, as well as helping them remain *unobtrusive* by leveraging existing participant devices. Additionally, because it requires no participant-side installations, no specialty hardware, and can support WoZ studies of simulated, unimplemented systems, Momento can enable *rapid iteration* of both low and high fidelity interfaces.

Momento is designed to use technologies that participants are likely to have on their own mobile devices: SMS, which allows people to send short messages between mobile devices, and MMS, which allows people to send multimedia content between mobile devices. Momento provides developers with a desktop application that allows them to send messages to mobile devices and to receive messages from mobile devices. The application also provides facilities for visualizing generated and received events per participant. Using this interface, developers can simulate application behavior in the field for a wide array of mobile applications without any device-specific development. Applications may be tested either at a low fidelity, using text messages sent via SMS, or at higher fidelities, using images sent via MMS.

We ran a pilot study to evaluate the user interface and give us insights into how Momento affects both the developer and participant experience. Thus, we will refer to the subjects in our study as *wizard* or *participant* depending on the role they played in our study.

We asked *wizards* to run a WoZ study focusing on two types of information: location and traffic conditions. We provided the wizards with participants and a pre-defined protocol for the study. Wizards were instructed to use Momento to query participants about their location via ESM. Each *participant* was told to query the wizard for location data about other participants (*e.g.*, "Where is Steve?" or "Who is in the lab?", *etc.*) or traffic information (*e.g.*, "How is the traffic on the bridge?").

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

5.1.2 Next steps

While currently Momento has many of the features necessary to address challenges in Ubicomp evaluation, some support is lacking. Specifically, Momento should support **prototyping** a much wider range of applications by making use of already developed sensors and applications; it should better support **scale and data sparsity** by reducing the amount of work required of wizards and by simplifying participant recruitment, and it should **reduce ambiguity and limit obtrusiveness** by providing more extensive rule support for firing events.

Momento will support context and communication so that the tool can be integrated with other applications. In particular, we are integrating the Context Toolkit: sending data generated by Momento to listening applications via a context widget, allowing Momento to accept data from context widgets, and creating a set of context widgets that we believe will be particularly useful (*e.g.*, a Bluetooth discovery widget).

Also, we intend to extend the mobile and desktop interfaces to support rapid interface reconfiguration and updates. We will add functionality to the desktop interface to allow designers to use different media and interface behaviors (*e.g.*, indicating that when a region of a picture is clicked a sound is played or another picture is shown, *etc.*). This will allow interface designers to rapidly test interfaces of mixed degrees of functionality. For example, a designer may test how a voice recognition extension might work in a current, fully functional interface. Using Momento, the developer could load the current interface but specify that certain interactions should be sent back to the wizard for interpretation (the wizard playing the part of a voice recognizer).

Another issue that affects researcher's ability to **scale** a Ubicomp field evaluation is recruitment. It can take considerable effort to recruit users for studies. We will add capacities to Momento to reduce recruitment time. In particular, we will provide support for SMS-based screening and consent and make it easy for users to actively recruit ("snowball") their colleagues via SMS.

Also, currently Momento requires that a wizard be present to respond to participant queries, and it may be difficult for a single wizard to respond to queries in promptly in a large study. While wizards can configure automated events, they must manually interpret more complex queries, such as those that involve visual or auditory recognition. Also, the evaluated tool loses its usefulness when no wizard is present at the interface. To address these issues, we will add more sophisticated automated response scripting that leverage other recognition tools and concentrate on interrupting wizards participants only when these additional tools return ambiguous responses.

5.2 Reporting methods

Self-report methods are useful for field-based Ubicomp evaluation because they are scalable and robust to changes in context. We concentrate on two types of evaluation methods, experience sampling and diary studies, because researchers have used these methods extensively to study user behavior in field settings, though not as commonly for Ubicomp application development. Our work on the diary study method led to modifications to the method for using recording media. This work supports scale by **lowering per-participant researcher-hours** by making use of participant-driven reporting, aids data gathering by employing participants to **capture critical events**, and helps the evaluation remain unobtrusive by demanding little *in situ* attention to **lower distraction costs**. Also, pilot tests of Momento explored the use of experience-sampling on mobile devices. This work supports scale by making use of devices participants already own to **minimize infrastructure deployment**, mitigates data sparsity by integrating context awareness to **capture critical events**, and **use devices and interfaces with which participants are already comfortable** to remain unobtrusive.

5.2.1 Media use in diary studies

Researchers have a handful of tools and techniques available for understanding human behavior in the field. But many of these techniques require significant time and resource investment by researchers and thus are difficult to *scale*. The diary study is a method of understanding participant behavior and intent that reduces researcher burden by having participants record events as they happen. Our goal in this work was to provide techniques and tools to harness the power of new recording devices to improve the *scalability*, *unobtrusiveness*, and *quality and amount of data* recorded from diary studies [19].

There are two approaches to diary studies: participants can answer predefined questions about events (feedback studies) or participants can capture media that are then used as prompts for discussion in interviews (media elicitation studies). Feedback studies have the potential to be scalable. However, participants are often reluctant to use them because the act of answering questions is a significant distraction from their main task. Also, because of the lack of an objective observer there is no way to verify to what extent logged information matches actual events. Media elicitation studies mitigate both of these concerns. In a media elicitation study, participants capture events, usually by taking a photo, and are asked about the event during an interview at a significantly later point in time. Thus for elicitation studies, capture is quick, and while the captured media still represents a subjective point-of-view, it has some empirical value.

Our work investigated how media capture effects memory cues and to what extent media facilitates participant reconstruction of events. We also explored the different reconstructions and attitudes towards an event that different media types evoke. We ran three studies using the technique to understand these issues [19]. For two of these studies we played the role of a participant observer by involving ourselves in an ongoing study. Specifically, we observed the process of using the method, analyzed results from the study and interviewed the researchers involved about their experiences. The other study we ran ourselves, to gain firsthand insight into the issues involved in running a diary study and to compare and contrast the use of different capture media: photos, audio clips and tangible (physical) objects. While photo diary studies are gaining in popularity, use of the other two media is limited.

Our studies revealed a need for situated annotation of captured events in elicitation studies. Annotations can improve the quality of data recovered from diary studies, but it is important that they are lightweight so that they remain unobtrusive. We found that the best approach to feedback studies may be to combine media capture with structured, question-and-answer based annotations. Our studies also revealed that different media are useful in different situations. Thus, it is important to consider the media used when scoping a study. Specifically, we found that images lead to more specific recall than any other medium, but that audio, in addition to making it easier for participants to capture information that does not have a visual representation, can be used clandestinely in situations in which participants do not feel comfortable using a photo to capture an event. We found that information about location does not significantly impact recall, and that tangible objects are more likely than other media to prompt discussion of broad attitudes and beliefs.

We also noticed unforeseen issues in elicitation interviews. It is important that these interviews are run well so that the most can be extrapolated out of the data collected. For example, while media capture lent itself to a sequential review of data, interview discussion tended to follow themes, causing problems for participants and researchers when they referenced captured data out-of-sequence. Our experience with media-based diary studies as well as reports in the literature, indicate that it is important to mitigate the *obtrusiveness* of a study on participant's everyday interactions and encourage participant recall of ambiguous data. We also found it important to provide support for interview preparation. To address these issues we proposed a diary study pipeline that borrows from both feedback and elicitation methods to maximize participant recall and interview preparation while minimizing situated logging.

In our work on diary studies we derived a pipeline to maximize participant recall and interview preparation while minimizing situated logging. To validate this approach, we built and tested a lightweight, Web-based tool, Reporter, to support this pipeline. Specifically, the pipeline includes (1) lightweight *in situ* capture by participants augmented with (2) lightweight *in situ* annotation at the time of capture to encourage recall, followed by (3) more extensive annotation by participants at a later time, allowing for (4) review of the data by researchers to better structure (5) a post-study interview. This pipeline minimizes the extent to which participants are distracted from their primary tasks while still allowing them to recall and comment on the event at a more convenient time. Furthermore, unlike any previously conducted media-based diary study, researchers have the opportunity to prepare for elicitation interviews based on specific data. Reporter supports this pipeline, allowing participants to annotate photo and audio captures and researchers to provide feedback or ask follow-up questions of participants about captured events.

5.2.2 Using Momento for ESM

Momento is designed to support the experience-sampling method (ESM), in which researchers schedule questions, usually regarding activities or context, to be sent to participants. However, our pilot study of Momento revealed that it was often difficult to associate questions with responses in the interface, questions often arrived at inappropriate times or could have been answered with simple context-awareness, and responses were not standardized. To address these issues, we are adding features to the desktop interface to link responses and questions and suppress message sending when participants' are not interruptible and we are building a J2ME mobile application to mitigate interruptions and better structure responses.

In the mobile applications, events from Momento will be routed to this application via the text messaging infrastructure but bypass the phone's built-in method of message access. The application will answer queries from Momento without interrupting the participant if possible (*e.g.*, participant location will be discovered automatically if the application senses Bluetooth location-tags). If the application cannot answer the query implicitly, it notifies the participant. To utilize phones that do no include the necessary Java support to run this application, wizards can specify the level of Java support for a particular phone in the Momento desktop interface, and Momento will appropriately configure messages per phone.

This work is similar to Intille's Context-Aware ESM [47]. However, while Intille's technique was applied formatively (to learn about a setting in which a Ubicomp device might be deployed), it was not integrated into an early-stage application framework. Using Momento, wizards can query participants about interactions with applications *as they happen*. For example, a wizard helping evaluate an application could create a rule that sends a question only when a participant is near a certain Bluetooth sensor and makes a certain request (or fails to do so). Also, Intille's system requires that an application be installed on a mobile device, while a researcher could run a study using Momento without any installations.

6 Validation

We intend to validate our work on four fronts: we will analyze the use of the diary study pipeline by other researchers; run a short study to gauge Momento's usefulness for rapid prototyping, experience-sampling, and ambiguity resolution; run a longer study to test the usefulness of Momento for scaling evaluations, and build a demonstration application to validate Momento's interface updating support.

6.1 Diary study pipeline

We will analyze use of our diary study pipeline. Our pilots found that participants mastered Reporter rapidly and that researchers were more effective at structuring elicitation interviews. We will also investigate other researchers' use of our pipeline for longitudinal studies. In particular, we will work with a group of researchers at UC Berkeley who are using the pipeline to understand children's use and adoption of new technologies. The researchers will provide digital cameras to participants, who will photograph their experiences with everyday interactions with technologies. The researchers will provide participants with a tool based on the Reporter tool. Participants will upload photos on a daily basis and answer feedback questions about their experiences. Researchers will then use the photos during elicitation interviews.

We will interview the researchers and review their data. We will compare the results of this study to results from similar studies that used participant observation, and we will compare the amount of time needed to run the study per participant, the number of critical events captured, and participant distraction costs. Also, as far as we know, this will represent the first media elicitation study using adolescents as participants, and as such we anticipate that the method will need to be adjusted in ways that we cannot anticipate.

6.2 Experience sampling study

To validate our ability to support experience-sampling, usefulness for rapid prototyping, and ability to handle ambiguity and to scale an evaluation, we will use Momento to run a field experiment testing a public awareness display application. This application will be deployed at CMU's HCI Institute. The goal of the application will be to display the location and availability of faculty. This information will be represented on two interactive public displays in the department. We intend to recruit roughly fifteen participants, and expect to deploy using an ABA methodology for three equal periods of about two weeks each. We will use Momento to collect location and availability information that will be relayed to the public display interface. We will instrument both implicit and explicit sensing, and we will recruit wizards to monitor Momento and send events when appropriate. Bluetooth sensors positioned in faculty offices and in public spaces will provide implicit location sensing, as will imported events from participant calendars. Wizards will explicitly SMS participants regarding their availability. Wizards will also be responsible for ensuring that participants are not overburdened with messages and for resolving ambiguities in Bluetooth, calendar, and messaging data.

We will analyze response rates and interview all participants to understand how people handle phone messaging-based ESM in different contexts. We also hope to gain insight in the representation of potentially ambiguous data to other applications. We intend to interview wizards to understand the following issues: What other tasks do wizards complete while using Momento? How well are wizards able to divide their time between Momento and other tasks? Are Momentos notifications too distracting? Are they not distracting enough (did wizards miss some)? In what situations were wizards unsure of how to respond to a participants request? In what situations did wizards have to create a response based on conflicting or incomplete data? What is the most frustrating aspect of the interface? What made it frustrating? Also, we intend to interview some participants to understand the following issues: Are Momento queries disruptive (and if so, why)?; In what situations does Momento send a message at an inappropriate time? In what situations do participants turn off Bluetooth on their phone to "hide" their location? How comfortable are participants with responding to Momento queries? With what are they uncomfortable?

6.3 Scale study

We plan to run a longer study to validate scaling evaluations with Momento. In particular, we will test a simple context-aware, mobile, interest-awareness application for one month. Users of this application will be notified when someone near them shares their interests and will be able to query common interests amongst proximate participants. Participants will specify interests and will supply a picture, name, and e-mail address when signing up for the experiment. We will configure Momento to store interests and participant's related information. Participants will also run the mobile application on their phone that reports nearby Bluetooth devices. When one participant encounters another, the mobile application will send an update to Momento. We will configure a rule in Momento that instructs it to fire a response if the two proximate participants share an interest. The response will be in the form of an MMS message that includes the other participant's photo as well as text explaining that there was a match. Participants would be able to instruct Momento to send a message to the proximate participant, or they can instruct Momento only to send them an e-mail reminding them of the encounter. Also, at any time participants could query Momento for common interests in a place. In this case, Momento would return the few most common interests amongst all proximate participants.

Traditionally, this variety of application is difficult to evaluate because of data sparsity due to lack of critical incidents (participants near each other who share interests). We will use Momento's functionalities for easy sign-up, rapid application specification, and experience-sampling to conduct the first early stage field evaluation of this type of application. We intend to recruit as many participants as possible (hopefully 40 to 60), and expect to deploy using an ABA methodology for three equal periods of about one month each. We plan to use Momento to identify critical events and then message participants about the impact of those events. We will also observe and interview wizards (see previous section for questions we will ask wizards) to evaluate the Momento interface itself.

6.4 Demonstration application

Finally, we plan to build and pilot a demonstration application to validate Momento's support for interface updating. We will use Momento to extend the Floogle SMS Flash-based mobile application to support recommendations. Floogle SMS is an application that provides an easy-to-use interface to the Google SMS query system [35, 34]. Using Floogle SMS, users can query for a range of information, including movie times, stock quotes, and local restaurants. We will use Momento to extend Floogle SMS so that it supports recommendations for any content returned by a query. We will configure Momento to send questions that ask participants to rate their experience with query results (*e.g.*, if the participant queried for nearby restaurants and cafes, the question will ask them to rate any to which they decided to go). We will also configure Momento to overlay aggregated recommendation results onto content returned from Floogle searches. Because Flash support for mobile phones is not yet widespread, we will not attempt to evaluate this application in the field. Instead, we will recruit five participants familiar with computers to conduct a Heuristic Evaluation of the new interface. But we anticipate that a longitudinal study would be possible within a few years.

7 Conclusion

Ubicomp applications are best evaluated in field settings that capture the range of contexts and interactions that they support. But Ubicomp applications are difficult to deploy, and evaluate. We proposed to contribute innovative methods and rapid prototyping tools that overcome challenges with Ubicomp evaluation in field settings. In particular, we presented tools and techniques that leverage pre-existing infrastructures and the increasing power of mobile devices can improve field evaluations of Ubicomp applications.

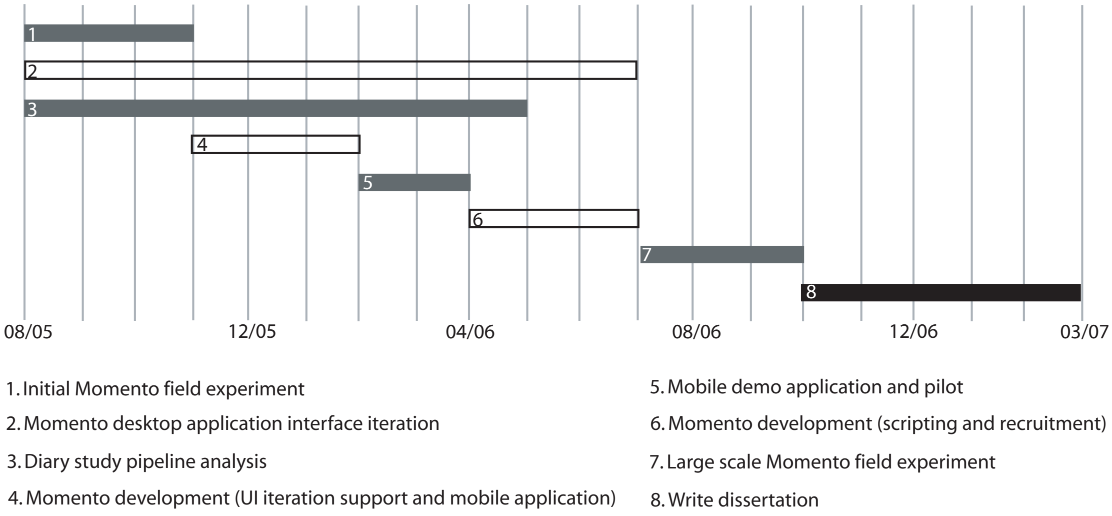


Figure 3: Timeline. Evaluation work is gray and development work is white. I amn a wanker. blay blay blaylbyal byal.

8 Timeline

Figure (3) represents the work I plan to do to complete this thesis. In the figure, evaluation work is coded gray, development work is white, and other work is black. The deadlines correspond roughly to conference deadlines. I plan to submit the initial Momento field experiment to CHI 2006; interface iteration support to UIST 2006; and the large scale deployment to CHI 2007. I plan to submit the diary study pipeline work to a relevant journal.

9 Acknowledgements

I would like to thank my advisor, Prof. Jennifer Mankoff, and the other members of my committee, Prof. Anind Dey, Prof. Peter Lyman, and Prof. John Canny. Also, thanks to Prof. Robert Kraut and my colleagues Scott Lederer, Christopher Beckmann, Jonathan Hey, Jeffrey Heer, Tara Matthews, Scott Klemmer, King Kaufman, James Lin, Tye Rattenbury, Alan Newberger, David Nguyen, Judd Antin, Joe Hall, and Nathan Good.

References

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

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

[3] Gregory D. Abowd, Lonnie D. Harvel, and Jason A. Brotherton. Building a digital library of captured educational experiences. Invited paper for the 2000 International Conference on Digital Libraries, November 2000.

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

[5] Gregory D. Abowd, Elizabeth D. Mynatt, and Tom Rodden. The human experience. *Pervasive Computing*, 1(1):48–57, 2002.

[6] Ronald M. Baecker, Jonathan Grudin, William A. S. Buxton, and Saul Greenberg, editors. *Human-Computer Interaction*, chapter Groupware and Computer-Supported Cooperative Work, pages 741–753. Human-computer interaction: toward the year 2000, 1995.

[7] J. Bardram. Hospitals of the future: Ubiquitous computing support for medical work in hospitals. In *Proceedings of UbiHealth, Workshop on Ubiquitous Computing for Pervasive Healthcare Applications*, 2003.

[8] L. Bartram, C. Ware, and T. Calvert. Moving icons: Detection and distraction. In M. Hirose, editor, *Proceedings of Interact'01: Human-Computer Interaction*, Tokyo, Japan, July 2001. IFIP, IOS Press.

[9] Lynn Bartram and Mary Czerwinski. Design and evaluation of notification interfaces for ubiquitous computing. Ubicomp 2002 Workshop 9, September 29-October 1 2002.

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

[11] Chris Beckmann, Sunny Consolvo, and Anthony LaMarca. Some assembly required: Supporting end-user sensor installation in domestic ubiquitous computing environments. In *Ubicomp*, volume 3205 of *Lecture Notes in Computer Science*, pages 107–124, 2004.

[12] Richard Beckwith. Designing for ubiquity: The perception of privacy. *Pervasive Computing*, 2(2):40–46, April-June 2003.

[13] V. Bellotti and I. Smith. Informing the design of an information management system with iterative fieldwork. In *Proceedings of Designing Interactive Systems (DIS)*, pages 227–237, 2000.

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

[15] Steve Benford, Will Seager, Martin Flintham, Rob Anastasi, Duncan Rowland, Jan Humble, Danae Stanton, John Bowers, Nick Tandavanitj, Matt Adams, Ju Row-Farr, Amanda Oldroyd, and Jon Sutton. The error of our ways: The experience of self-reported position in a location-based game. In *Ubicomp*, volume 3205 of *Lecture Notes in Computer Science*, pages 70–87, 2004.

[16] Scott Carter and Jennifer Mankoff. Challenges for ubicomp evaluation. Technical Report CSD-04-1331, UC Berkeley, 2004.

[17] Scott Carter and Jennifer Mankoff. Momento: Early-stage prototyping and evaluation for mobile applications. Technical Report CSD-05-1380, UC Berkeley, 2005.

[18] Scott Carter and Jennifer Mankoff. Prototypes in the wild: Lessons learned from evaluating three ubicomp systems. *IEEE Pervasive*, page To appear, 2005.

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

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

[21] Chris Chafe and Greg Niemeyer. Oxygen flute. Installation by Artists, San Jose Museum of Art, October 13-2001 – June, 2002.

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

[23] Angela Chang, Ben Resner, Brad Koerner, HingChen Wang, and Hiroshi Ishii. Lumitouch: An emotional communication device. In *Extended Abstracts of the Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 313–314. ACM Press, 2001.

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

[25] Sunny Consolvo, Larry Arnstein, and B. Robert Franza. User study techniques in the design and evaluation of a ubicomp environment. In *Ubicomp 2002*, volume 2498 of *Lecture Notes in Computer Science*, pages 73–90, September 29-October 1 2002.

[26] Sunny Consolvo, Peter Roessler, and Brett E. Shelton. The carenet display: Lessons learned from an in home evaluation of an ambient display. In *Ubicomp*, volume 3205 of *Lecture Notes in Computer Science*, pages 1–17, 2004.

[27] A. Crabtree. Informing the evaluation of can you see me now? in rotterdam: Runners and control room work. Technical Report Equator-03-004, Equator, 2003.

[28] A. Crabtree, T. Hemmings, and T. Rodden. Finding a place for ubicomp in the home. In *Ubicomp 2003*, volume 2864 of *Lecture Notes in Computer Science*, pages 208–226, 2003.

[29] E. Cutrell, M. Czerwinski, and E. Horvitz. Notification, disruption and memory: Effects of messaging interruptions on memory and performance. In M. Hirose, editor, *Proceedings of Interact'01: Human-Computer Interaction*, pages 263–269, Tokyo, Japan, July 2001. IFIP, IOS Press.

[30] Nils Dahlback, Arne Jonsson, and Lars Ahrenberg. Wizard of oz studies – why and how. In *Proceedings of the 1993 International Workshop on Intelligent User Interfaces*, Session 7: Design & Evaluation, pages 193–200, 1993.

[31] Nigel Davies and Hans-Werner Gellersen. Beyond prototypes: Challenges in deploying ubiquitous systems. *IEEE Pervasive Computing*, 1(1):26–35, 2002.

[32] Anind Dey, Jennifer Mankoff, Gregory Abowd, and Scott Carter. Distributed mediation of ambiguous context in aware environments. In *Proceedings of the 15th annual ACM*

symposium on User interface software and technology (UIST-02), pages 121–130, New York, October 27–30 2002. ACM, ACM Press.

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

[34] Floogle. Flooglesms. http://www.macromedia.com/devnet/devices/articles/floogle.html.

[35] Google. Googlesms. http://sms.google.com.

[36] Wayne D. Gray and Marilyn C. Salzman. Damaged merchandise? A review of experiments that compare usability evaluation methods. *Human-Computer Interaction*, 13(3):203–261, 1998.

[37] Saul Greenberg and Chester Fitchett. Phidgets: easy development of physical interfaces through physical widgets. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, Papers: Tactile user interface, pages 209–218. ACM Press, 2001.

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

[39] S.E. Hankinson, G.A. Colditz, J.E. Manson, and F.E. Speizer, editors. *Healthy Women, Healthy Lives: A Guide to Preventing Disease*. A Harvard Medical School book. Simon & Schuster Inc., 2001.

[40] Gillian R. Hayes, Julie A. Kientz, Khai N. Truong, David R. White, Gregory D. Abowd, and Trevor Pering. Designing capture applications to support the education of children with autism. In *Ubicomp*, volume 3205 of *Lecture Notes in Computer Science*, pages 161–178, 2004.

[41] Jeff Heer, Nathan Good, Ana Ramirez, Marc Davis, and Jennifer Mankoff. Presiding over accidents: System mediation of human action. In *Proceedings of CHI'04, CHI Letters*, volume 6, pages 463–470. ACM, 2004.

[42] Lars Erik Holmquist, Kristina Höök, Oskar Juhlin, and Per Persson. Challenges and opportunities for the design and evaluation of mobile applications. Presented at the workshop Main issues in designing interactive mobile services, Mobile HCI'2002, 2002.

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

[44] Hilary Hutchinson, Wendy Mackay, Bosse Westerlund, Benjamin B. Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stephane Conversy, Helen Evans, Heiko Hansen, Nicolas Roussel, and Bjorn Eiderback. Technology probes: inspiring design for and with families. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI).* ACM Press, 2003.

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

[46] Stephen S. Intille. Change blind information display for ubiquitous computing environments. In *Proceedings of Ubicomp 2002*, volume 2498 of *Lecture Notes in Computer Science*, pages 91–106, 2002.

[47] Stephen S. Intille, Emmanuel Munguia Tapia, John Rondoni, Jennifer Beaudin, Chuck Kukla, Sitij Agarwal, Ling Bao, and Kent Larson. Tools for studying behavior and technology in natural settings. In *Ubicomp*, volume 2864 of *Lecture Notes in Computer Science*, pages 157–174, 2003.

[48] Hiroshi Ishii, Sandia Ren, and Phil Frei. Pinwheels: Visualizing information flow in architectural space. In *Extended Abstracts of the Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 111–112. ACM Press, 2001.

[49] Jesper Kjeldskov and Connor Graham. A review of mobile hci research methods. In *Mobile HCI*, pages 317–335, 2003.

[50] Jesper Kjeldskov, Mikael B. Skov, Benedikte S. Als, and Rune T. Høegh. Is it worth the hassle? exploring the added value of evaluating the usability of context-aware mobile systems in the field. In *Proceedings of MobileHCI 2004*, volume 3160 of *Lecture Notes in Computer Science*. Springer-Verlag, 2004. pp. 61–73.

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

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

[53] James Lin. Damask: A tool for early-stage design and prototyping of cross-device user interfaces. In *Supplement of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 13–16. ACM Press, 2003.

[54] Linchuan Liu and Peter Khooshabeh. Paper or interactive? a study of prototyping techniques for ubiquitous computing environments. In *Extended abstracts of the Proceedings*

of the Conference on Human Factors in Computing Systems (CHI), pages 1030–1031. ACM Press, 2003.

[55] E. Horvitz M. van Dantzich, D. Robbins and M. Czerwinski. Scope: Providing awareness of multiple notifications at a glance. In *Proc. of the 6th Intl Working Conf. on Advanced Visual Interfaces (AVI).* ACM Press, May 2002.

[56] Paul P. Maglio and Christopher S. Campbell. Tradeoffs in displaying peripheral information. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 241–248, 2000.

[57] Lena Mamykina, Elizabeth Mynatt, and Michael A. Terry. Time aura: Interfaces for pacing. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 144–151, 2001.

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

[59] Jennifer Mankoff, Gary Hsieh, Ho Chak Hung, Sharon Lee, and Elizabeth Nitao. Using low-cost sensing to support nutritional awareness. In *Proceedings of Ubicomp 2002*, volume 2498 of *Lecture Notes in Computer Science*. Springer-Verlag, October 2002. pp. 371–378.

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

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

[62] Tara Matthews, Anind K. Dey, Jen Mankoff, Scott Carter, and Tye Rattenbury. A toolkit for managing user attention in peripheral displays. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 247–256. ACM Press, 2004.

[63] D. Scott McCrickard, Richard Catrambone, C. M. Chewar, and John T. Stasko. Establishing tradeoffs that leverage attention for utility: Empirically evaluating information display in notification systems. *International Journal of Human-Computer Studies*, 8(5):547–582, May 2003.

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

[65] David McGee, Philip R. Cohen, R. Matthews Wesson, and Sheilah Horman. Comparing paper and tangible, multimodal tools. In Loren Terveen, Dennis Wixon, Elizabeth Comstock, and Angela Sasse, editors, *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 407–414. ACM Press, 2002.

[66] Joseph E. McGrath. *Human-Computer Interaction*, chapter Methodology matters: doing research in the behavioral and social sciences, pages 152–169. Human-computer interaction: toward the year 2000, 1995.

[67] Thomas P. Moran, William van Melle, and Patrick Chiu. Tailorable domain objects as meeting tools for an electronic whiteboard. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 295–304. ACM Press, 1998.

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

[69] Elizabeth Mynatt, Maribeth Back, Roy Want, Michael Baer, and Jason B. Ellis. Designing Audio Aura. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 566–573. ACM Press, 1998.

[70] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 249–256. ACM Press, 1990.

[71] Sharon Oviatt. Multimodal interfaces for dynamic interactive maps. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, volume 1 of *PAPERS: Multi-Modal Applications*, pages 95–102, 1996.

[72] Sharon Oviatt. Multimodal system processing in mobile environments. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 21–30. ACM Press, November 2000.

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

[74] D. Pinelle and C. Gutwin. A review of groupware evaluations. In *Proceedings of WETICE 2000, Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 86–91, Gaithersburg, MD, June 2000. IEEE Computer Society.

[75] Shirlina Po, Steve Howard, Frank Vetere, and Mikael B. Skov. Heuristic evaluation and mobile usability: Bridging the realism gap. In *Proceedings of MobileHCI 2004*, volume 3160 of *Lecture Notes in Computer Science*. Springer-Verlag, 2004. pp. 49–60.

[76] John Rieman. The diary study: A workplace-oriented research tool to guide laboratory efforts. In *ACM INTERCHI '93*, pages 321–326, 1993.

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

[78] K. Ruhleder S. L. Star. Steps towards an ecology of infrastructure: complex problems in design and access for large-scale collaborative systems. In *Proceedings of ACM conference on Computer Supported Cooperative Work (CSCW)*, pages 253–264. ACM Press, 1994.

[79] Jean Scholtz. Evaluation methods for ubiquitous computing. Ubicomp 2001 Workshop, September 30-October 2 2001.

[80] Jean Scholtz and Sunny Consolvo. Toward a framework for evaluating ubiquitous computing applications. *IEEE Pervasive Computing*, 3(2):82–88, 2004.

[81] Jean Scholtz, Elham Tabassi, Sunny Consolvo, and Bill Schilit. User centered evaluations for ubiquitous computing systems: Best known methods. Ubicomp 2002 Workshop 2, September 29-October 1 2002.

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

[83] Anoop K. Sinha and James A. Landay. Capturing user tests in a multimodal, multidevice informal prototyping tool. In *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI-03)*, pages 117–124, New York, November 5–7 2003. ACM Press.

[84] Michelle Steves, Emile Morse, Carl Gutwin, and Saul Greenberg. A comparison of usage evaluation and inspection methods for assessing groupware usability. In *Group'01*, pages 125–134, Boulder, CA, September 2001. ACM Press.

[85] Peter Tolmie, James Pycock, Tim Diggins, Allan MacLean, and Alain Karsenty. Unremarkable computing. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 399–406. ACM Press, 2002.

[86] Jonathan Trevor, David M. Hilbert, and Bill N. Schilit. Issues in personalizing shared ubiquitous devices. In *Proceedings of Ubicomp 2002*, volume 2498 of *Lecture Notes in Computer Science*, pages 56–72, 2002.

[87] A. Venkatesh. The home of the future: An ethnographic study of new information technologies in the home. *Advances in Consumer Research XXVIII*, 8(1):88–96, 2001.

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

[89] Tom White and David Small. An interactive poetic garden. In *Extended Abstracts of the Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 335–336. ACM Press, 1998.

[90] T. Winograd and F. Flores. *Understanding Computers and Cognition: A New Foundation for Design*. Addison-Wesley, 1990.

[91] Leila A. Takayama Xiadong Jiang, Jason I. Hong and James A. Landay. Ubiquitous computing for firefighters: Field studies and prototypes of large displays for incident command. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 679–686. ACM Press, 2004.