Challenges for Ubicomp Evaluation

Scott Carter and Jennifer Mankoff
EECS Department
UC Berkeley
sacarter,jmankoff@cs.berkeley.edu

June 7, 2004

Abstract

Technology in different forms is available ubiquitously through much of the world. The study of Ubiquitous Computing (Ubicomp) is concerned with enabling a future in which the most useful applications of such technology are feasible to build, and pleasing to use. But what is useful? What is usable? What do people actually need? These questions are only beginning to be answered partly because Ubicomp systems are more difficult to evaluate, particularly at the early stages of design, than desktop applications. This difficulty is due to issues like scale and ambiguity, and a tendency to apply Ubicomp in ongoing, daily life settings unlike task and work oriented desktop systems. This paper presents a case study of three Ubicomp systems that were evaluated at multiple stages of their design. In each case, we briefly describe the application and evaluation, and then present a set of lessons that we learned regarding the evaluation techniques we used. Our goal is to better understand how evaluation techniques need to evolve for the Ubicomp domain. Based on the lessons that we learned, we suggest four challenges for evaluation.

1 Introduction

Technology in different forms is available ubiquitously through much of the world. The study of Ubiquitous Computing is concerned with enabling a future in which the most useful applications of such technology are feasible to build, and pleasing to use. Feasibility depends on the availability of network connectivity and data, of sensors and algorithms for interpreting the data they produce, and of tools with which to ease the building of applications. The degree to which an application is pleasing and useful depends on the ability of application designers to understand and meet the needs of the users of ubiquitous computing (Ubicomp) applications. User needs are best understood with the help of evaluation techniques.

One of the hardest problems that application developers face today is the difficulty of evaluating ubiquitous computing applications. Because of this, the evaluation of Ubicomp systems is an active area of discussion and research [16, 41, 63, 65, 68]. Evaluation is crucial at all stages of design, and the best designs include evaluations that involve users in the design process repeatedly throughout a series of design iterations.

Ubiquitous Computing (Ubicomp) applications are applications that are embedded seamlessly and ubiquitously into our everyday lives. A commercial example is the cellphone. Another example is the vision of technology suggested in the recent movie, Minority Report, from the advertisements that recognize and activate as a person walks by, to the home that responds to voice commands. Often these systems depend on a large number of sensors, must operate across a variety of handheld, desktop, and wall-mounted devices, or must operate over a large number of people or across a long period of time.

We argue in this article that, despite their popularity in the Graphical User Interface (GUI) world [71], early-stage, discount evaluation techniques are particularly lacking from the suite of evaluation methods currently used by and available to Ubicomp designers. By this, we mean techniques that are applied after

requirements gathering is completed and an initial application idea exists, but before a working implementation is completed. Examples of discount and early-stage evaluation techniques include paper prototyping and heuristic evaluation. Although non-discount techniques may also be applied in some cases, discount techniques are particularly well suited to early-stage design because, like the prototypes being evaluated, they typically require little effort, time, and money, to be applied. Thus, they help to make iterative design feasible, allowing designers to get more, and more frequent, feedback about the potential of their designs for success.

This paper presents a case study of three Ubicomp systems that were evaluated at multiple stages of their design. In each case, we briefly describe the application and evaluation, and then present a set of lessons that we learned regarding the evaluation techniques we used. Our goal is to better understand how evaluation techniques need to evolve for the Ubicomp domain. Based on the lessons that we learned, we suggest four challenges for evaluation.

1.1 Overview

Section 2 gives a review of some of the existing work in Ubicomp evaluation. In Section 3, we describe three case studies of our own experiences with Ubicomp evaluation, discussing what we learned about different evaluation techniques in each case. The first was a paper-based prototype study of PALplates, active elements placed around an office in locations of need. The second was a study of a prototype of a nutritional advice system deployed in domestic settings. The third, our keystone study, was a full deployment of hebb, a system for monitoring and encouraging cross-community relations. Hebb was evaluated 5 times, not counting formative research, over the course of 20 months during our iterative design process. We conclude with a discussion leading to four major challenges for Ubicomp evaluation.

2 Background

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 [63, 65, 6]. Additionally, recent articles have remarked on the difficulties and challenges facing Ubicomp Evaluation [27, 4]. 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, 50, 53, 77]). We begin by discussing 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, particularly at the early stages of system design.

2.1 Examples of systems where multiple evaluations occurred

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 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 [50], was 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 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 incident command centers, and then developed a third, improved display based on the results [77]. Their evaluation involved showing the prototypes to firefighters and asking for feedback. Mynatt *et al.* developed multiple designs of a digital family portrait, all before a working prototype was completed [53]. 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 Wizard-of-Oz evaluation [53].

In addition to these examples of iterative design, numerous developers have successfully used either existing formative evaluation or existing summative evaluation in the design of Ubicomp applications. Below, we give examples of both.

2.2 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. Areas of study include the home [17, 70], mobile document and information workers [10], hospitals [5], and many more arenas for Ubicomp deployment.

An early piece of work in early-stage evaluation of a non-working prototype was Oviatt's use of Wizard-of-Oz evaluation to test a multi-modal map application which combined speech and pen input in different ways [55] (Dählback was first to highlight the uses of Wizard-of-Oz systems in this domain [19].) Additionally, as mentioned above, Wizard-of-Oz evaluation has been applied to other Ubicomp systems [53]. Finally, Jiang *et al.* have used a form of user-based expert review to evaluate non-working systems [77].

2.3 Summative Evaluations

In contrast to formative evaluations, several varieties of summative evaluations have been applied to Ubicomp systems. Oviatt made use of a wearable computer to test the robustness of speech in unconstrained settings [56]. When combined with pen input, the system was remarkably effective in recognizing speech. A related study conducted by McGee *et al.* compared maps and post-it notes with a tangible multi-modal system [47]. This was done using a controlled, laboratory study format. Both studies demonstrated that, just as with desktop user interfaces, Ubicomp system design benefits from and is informed by thorough evaluation.

Consolvo *et al.* used Lag Sequential Analysis in a summative evaluation to gather quantitative data about how their system was used [16]. Although an interesting and informative technique, Lag Sequential Analysis requires hours of video to be coded by hand into different categories.

Beckwith interviewed and observed inhabitants of a nursing home to gather qualitative data regarding the inhabitants' perceptions of sensing technologies installed in the home [8]. The application of these standard qualitative methods is useful in uncovering social activities and perceptions of Ubicomp technologies, but they are time- and effort-intensive.

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. In the case of ambient displays, evaluations have rarely been done, and typically include few details or informal feedback, focusing on technology and design innovation (*e.g.* [14, 35, 52]).

Mamykina *et al.* [40] conducted a summative evaluation of an ambient display, showing it to be a useful tool. More informally, some ambient displays have been exhibited in museum or gallery settings where

they were used by hundreds of users but never tracked in detail (*e.g.* [12, 14, 34, 74]). The most common "finding" of exhibits and empirical studies of ambient displays is that users are interested in and excited by innovations in ambient displays.

Researchers investigating alerting displays have conducted more evaluations (*e.g.* [15, 38]). The alerting display community has also put significant effort into developing a deep understanding of interruptibility [28]; parallel and pre-attentive processing ([22, 75]); the attentional impact of different types of animations or other alerts (*e.g.* [7, 18, 33, 39, 45, 44, 46], and so on). However, these studies normally involve extensive lab-based testing rather than discount, real-world evaluations of actual systems.

2.4 Tailoring evaluation techniques to Ubicomp

Based on our survey, then, it appears that a limited set of existing techniques are being applied to early-stage Ubicomp evaluation, and in fact the main early-stage technique that has been applied successfully is Wizard-of-Oz evaluation [56, 53]. At the same time, a variety of techniques are being applied successfully at summative stage. This leads to two questions: Could other formative techniques be applied to the domain of Ubiquitous Computing? Could tools be developed that might make existing summative techniques more applicable at the early stages of design? Based on recent work described below, it seems clear that the answer to both questions is yes. Researchers are creating new techniques, and modifying existing techniques (or providing tools to support them) with great success.

Hutchinson *et al.* experimented with a new form of evaluation called Technology probes [29]. "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." These are a great example of how to use fairly lightweight prototypes to explore design ideas.

Another lightweight technique is Intille *et al.*'s automated method for asking users about what is going on during a moment in time captured by a video camera [32, 61]. This technique has the advantage of requiring little ongoing effort from end users or experimenters. However, it depends upon extensive infrastructure to capture images and send them to the user. This highlights the point that one may need a tool complex enough to be considered a Ubicomp environment in its own right to easily evaluate Ubicomp environments.

Chandler *et al.* developed a set of guidelines for paper prototyping of multi-modal applications [13]. This work was then expanded to support lightweight Wizard-of-Oz evaluation, using a technique called "Multimodal theater" [66]. 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 [37]. 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. Then again, we discuss a successful application of paper prototyping to Ubicomp application design later in this paper [43]. Clearly, additional investigation into appropriate uses of and support for paper prototyping is needed.

Mankoff *et al.* created a modified version of Heuristic Evaluation appropriate for peripheral displays [41]. Other researchers have created or adapted different techniques for use in formative Ubicomp evaluation settings, although none as lightweight as technology probes. Trevor *et al.* developed a comparative study methodology similar to a laboratory experiment [68]. 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* rather than for use, 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. As with other past work, this evaluation could not be considered lightweight. The deployment continued for several months.

2.5 Why is Ubicomp Evaluation so Difficult

In summary, a survey of existing work in Ubiquitous Computing and related fields shows that researchers and developers are very interested in evaluating Ubicomp applications, but typically focus either on studying users *before* design commences or on building working systems and then conducting heavyweight summative evaluations in either the field or the lab. Discount evaluations appropriate for use at the early stages of design are particularly rare. However, in recent years, some exciting and promising research in the evolution of lightweight techniques appropriate for Ubicomp evaluation has begun. Even so, the community believes that Ubicomp evaluation needs work [27, 4]. Why is Ubicomp evaluation, particularly *early-stage, discount* evaluation, so difficult?

We believe that one major issue is the difficulty of building Ubicomp systems. Heiner *et al.* (1999) reported spending as much as a year designing one Ubicomp application [26]. Additionally, Ubicomp applications are rarely used outside of a laboratory setting. This makes iteration difficult, and realistic studies of their use next to impossible. Even in the lab, it is hard to conduct a controlled a Ubicomp evaluation, because Ubicomp applications are generally designed to be integrated into complex applications and tasks. The infrastructure needed to conduct a study may constitute a Ubicomp environment in its own right. This means that, in the absence of evaluation techniques that do not depend on complete, working systems, developers currently must put significant effort into an application before testing it. Better early-phase evaluation techniques could help to address this issue.

Contrast this to a survey of desktop user centered design techniques, that found lightweight, discount evaluations to be half of the most commonly used methods [71]. Speed and cost were rated as two of the three top reasons to use these techniques. Although our survey was of research systems, while Vredenburg *et al.* looked at corporate designers, it is still surprising that so few similar, popular techniques are in use in the Ubicomp field. We attribute this to a lack of proven, effective, inexpensive evaluation techniques suitable for Ubicomp applications.

3 Three Case Studies

In this section, we present three case studies of evaluations we did ourselves of Ubicomp systems. The first, PALplates, was developed in 1997. The second, a nutritional advice system, is currently under development. 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.

3.1 PALplates

PALplates [43] 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.

3.1.1 Evaluation

Our evaluation of PALplates took the form of a paper prototype. Paper prototyping is traditionally done by sketching all of the dialog boxes, screens, and other interactive elements of a graphical user interface on paper. A human then simulates a computer's reactions as users execute pre-defined tasks using the paper interface [59]. This technique is effective, and so quick that we ask students to use it during one of the design iterations in our undergraduate interface design class. One reason it is so quick is that it requires no code development, but is able to provide early feedback on design ideas. It can also be used to involve end users in design. For example, Müller (1992) presented end users with a kit of user interface elements and asked for feedback about layout [51].

In our work, we posted paper interfaces in places of interest (office doors, common rooms, conference rooms) around an office place (an example is shown in Figure 1). People could write notes, or make requests on any paper interface. They could also make suggestions and reservations and see local news. Twice a day,

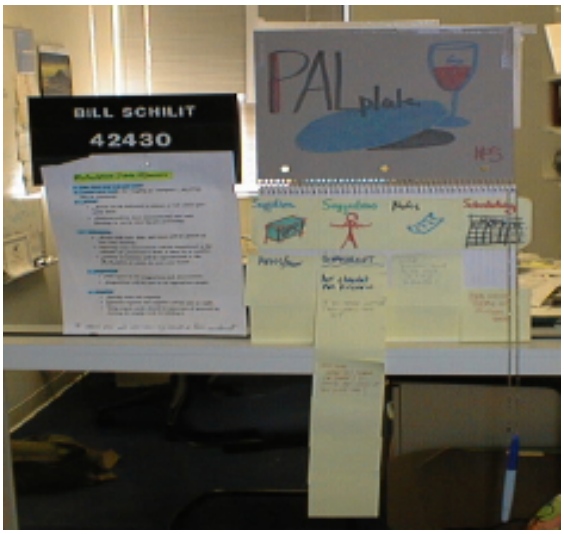


Figure 1: A paper protoytpe of the PALPlates system

a volunteer member of our network, "sneakernet", would visit each display and execute the requests. For example, a note might be picked up and delivered to the display for which it was addressed.

We found that even though PALplates were missing important features that would be present in a full-fledged application, people used them. PALplates in different locations were used for different tasks. For example, the display located in the kitchen was used mostly for discussion and ordering kitchen supplies. Although we expected functions to be closely tied to location, we also found "remote access" activity. For example, people reserved the meeting room from a PALplate located in the hallway sometimes. In addition, people were interested in using the PALplates to access their own private information such as Web documents and calendars.

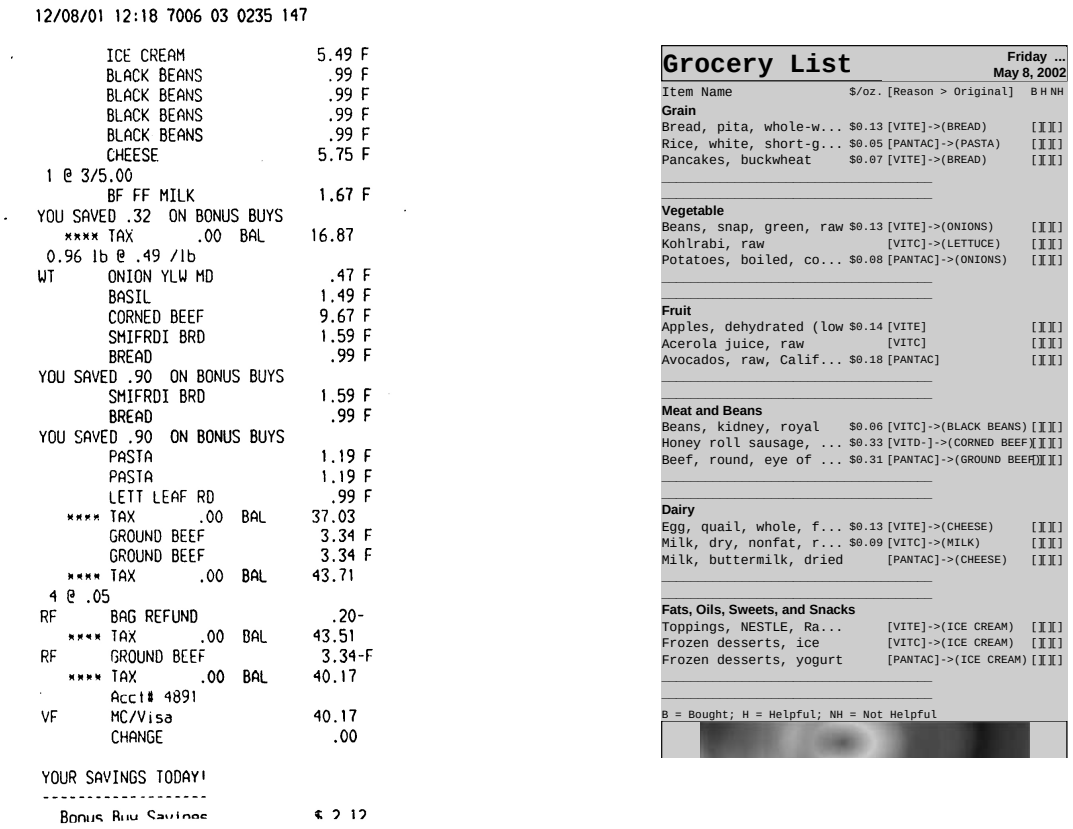
3.1.2 Lessons Learned

Did not scale across time – lack of responsiveness Despite our success with PALplates, we believe that paper prototyping has some serious limitations. In particular, as it stands, paper prototyping fails to adequately handle scale (Liu *et al.* arrived at a similar finding [37]). For example, in PALplates, we were could only update our displays twice daily. We were unable to simulate finer-grained control such as updating the printer status each time a job began or ended, or updating a map showing the current location of everyone in the building. Also, the limited nature of our updates precluded rapid-fire conversations between participants.

Errors rarely occurred With human "agents" it was difficult to demonstrate realistic errors in the PALplates project. Instead, things were perhaps too perfect. Given the number of volunteers needed to run the system, anything more sophisticated would have required too much training. Also, participants knew humans were involved, and expected fairly accurate, consistent behavior.

3.2 Nutrition Tracking

The prevalence of obesity (defined as a body mass index >= 30kg/m^2) increased from 12.0% in 1991 to 17.9% in 1998 in the United States [49]. This is a major public health concern since even a few extra pounds can increase the chance of developing chronic diseases such as cardiovascular disease and diabetes [24]. Healthy eating can help to reduce the occurrence of different health problems. Many people do not know exactly how



(a) **(b)**

Figure 2: **(a)** A receipt from a local grocery store. **(b)** A shopping list generated by our system, based on that receipt.

many servings of fruits, grains, vegetables, and fats they are eating, much less which nutrients are missing in their diet [31]. 7 out of 10 Americans believe that change is complicated [30]. Certainly, getting help from computers is complicated: existing solutions have the cumbersome requirement to enter by hand everything one eats.

We are working to build an application, shown in Figure 2, that uses inexpensive, low-impact sensing to collect data about what household members are purchasing and consuming, and uses simple yet persuasive techniques to suggest potential changes. The proposed system would gather data about purchasing habits when receipts are scanned in with a handheld scanner (this could be done when bills are being sorted at the end of the week). A shopping list, printed at the user's request, can provide annotated suggestions for slight changes in purchases. This portable piece of paper could provide suggestions at the most pertinent moment: when the user is making purchasing decisions. It can be used to encourage alternate, healthier purchases such as baked tortillas instead of chips, or wheat bread instead of white.

3.2.1 Evaluation

Currently, we have conducted surveys and interviews to guide the early stages of the design, and implemented portions of our proposed application [42]. We also developed and tested a paper prototype of this application. We iterated on our interface based on a series of informal studies in which we asked people to comment on

sketched designs. Finally, we deployed it for three weeks in the homes of three participants. Our deployment involved a two-week data gathering period, followed by three weeks of field use, followed by a follow-up interview.

Our results were mixed. Although we studied our users, talked with them, and tried portions of our interface out with them on paper, it was not until the deployment that we learned how poorly our interface met their needs. The problems uncovered during the deployment were certainly no more severe than a typical interface designer might find on the first iteration of a new system. But the amount of development effort, and the high overhead of the study are totally out of proportion with a first iteration. And at the end, our subjects told us that three weeks was only barely enough time for them to become comfortable with the system.

In retrospect, we could have predicted some of the problems that arose based on our interviews. The system was of limited use, for example, if someone does not use a shopping list or eats out often and shop at a particular store. An additional issue with our study was the difficulty of selecting subjects who met our ideal profile. Our interviews suggested that such subjects exist, and the people who volunteered for our study reported matching our needs. However, in practice we would have had to interview each of them in depth to determine if they were a good match. In fact, despite the fact that they used shopping lists, none of them used paper shopping lists *while shopping*, many of them bought bulk goods at a separate store, and most of them did not consider breakfast or lunch on the run eating out.

3.2.2 Lessons Learned

Could not measure ability to integrate unobtrusively The biggest lesson we learned from the Nutrition Tracking project was the difficulty of accurately assessing the problems and potential of a system that must integrate unobtrusively into a daily life activity. Our paper prototype failed to alert us that our system would not integrate well into shopping patterns.

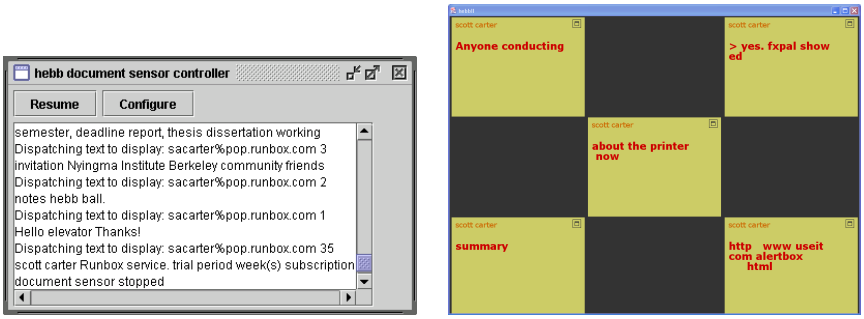
Did not scale across data – could not test recommendations Our Nutrition Tracking paper prototype also could not help us to assess the quality of our recommendations. They depended on having weeks of data that we could not gather and integrate without a working system.

Could not measure impact of errors The nutrition system was rife with ambiguity and error. The most useful information that came out of our deployment was that the recommendations were not of a high enough quality. Yet until the system was in use we could not easily tell what level of ambiguity was acceptible to users.

3.3 hebb

hebb is a Ubicomp system designed to capture and convey shared interests [11]. This system arose from a series of interviews that we conducted with members of six small working groups. From these interviews, we found that the benefits of collocation often do not extend beyond the group. Shared interests with nearby groups or individuals with similar interests go unnoticed, limiting the chance for communication and collaboration. 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. Each component registers with other components to receive semantically tagged data that component generates. The interest sensor generates picture, name and keyword events, corresponding to a picture of the user of the interest sensor, the user's name, keywords (generated from the algorithm described in the next section) as well as 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 documents keywords were recently displayed. A server on each component handles incoming requests by saving the IP



(a) (b)

Figure 3: 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.

address of the requesting component in an event-specific table. When a component generates an event, it sends it to all components listed in the table. Components, especially interest sensors, may go on and off line sporadically and may not be statically addressed. For this reason, a discovery server allows components to update and retrieve location information.

3.3.1 Evaluation

After attempting to recruit several different groups to use the system, we deployed it with two research groups in the same department. We were able to install the system with one group immediately, but the installation with the other group stalled and we eventually redeployed that installation with the author's research group.

Our first attempts to recruit groups to use the system were met with an unexpected resistance. We gave multiple presentations for multiple work groups but found it difficult both to find support for the system and to convey an appropriate conceptual model of the system. Some groups rejected the system outright because it would "increase e-mail spam" and that the public displays would "end up showing random ads." Further interviews with these groups showed that these responses seemed to reflect more an accepted ideology about technology than particular experiences. It should be noted that many of these groups have little previous experience with novel technologies.

Eventually, two academic research groups agreed to test the system. These two groups are in the same department and in the same building and working on similar topics. However, several floors separate the two groups: one group is located in the basement and the other on the top floor. Interviews revealed that these groups shared a considerable amount of work practice experiences but nonetheless did not communicate with one another. Thus, the two groups seemed a good match for the hebb system.

Installing this system required buy-in from the students and researchers in the lab as well as lab directors and computer networking staff. To encourage interest in the system, we gave several presentations to both groups and held several meetings with stakeholders regarding installation plans. As mentioned, one of these groups readily accepted the system. However, we found it difficult to convince the other group that the system was secure. In particular, a wireless network had to be installed in order to support private displays, but the lab director and students were concerned that a wireless network would invite attacks on their system to obtain private data from scientific studies that the lab relies heavily upon. Attempts to explain that their internal network and the wireless network could be separated were greeted with suspicion.

Furthermore, some users vehemently rejected the notion of an e-mail sensor. Again, explanations of the system's privacy assurances, that only high-level data would be transmitted and that course controls would be available to turn off the sensor, did little to placate these users. One even went so far as to say that he "would never let [the author] near [his] machine." It is notable that in the design of this system users widely

accepted the concept of an e-mail sensor.

After negotiations with this lab stalled, we deployed the system to two other academic research groups working on similar topics. The groups were spread across three different spaces: one group of three members was collocated while the other group of four was split between two spaces. This particular arrangement allowed us to determine how well the system supports intra- and inter-group awareness and communication. We deployed the interest sensor first for four weeks to establish common e-mail patterns and then deployed the rest of the system for another four weeks.

3.3.2 Lessons Learned

Anticipate Users to Adapt The primary lesson our deployments taught us was that user's adapted both their attitudes toward and their use of the technology over time. In particular, we found that users' attitudes toward the system changed as the system became less remarkable and more unobtrusive. For example, during the development stage and at the beginning of deployments, user attitudes towards the interest sensor were negative to skeptical. However, their attitude changed significantly over the course of the deployment.

Participant expectations for how to use displays evolve Participants tended to adapt their use and perception of peripheral output technologies over the course of the deployment. While at first users found it difficult to overcome using common technologies in new ways, they eventually adapted and felt comfortable with these new use modes.

Users have models of what kind and fidelity of information should be on given displays. In our deployment, LCD monitors with attached touchscreens served as public displays. Early in the deployment we asked about these design of these displays many users said that they were "overkill" for the task and wondered why higher fidelity content was not being displayed on them. In fact, in early deployments public displays at two sites were regularly switched to lab Web sites and new outlets (*e.g.*, CNN.com). However, later in the deployment users reneged on their former statements, identifying the displays less as pieces of high fidelity technology co-opted for low-fidelity use and more as a low-fidelity display only.

Also, we found participants adapted their use of the personal display. In particular, some participants used the PDA neither as a mobile device, as we had expected, nor as a display that they could monitor peripherally from their desktop. Rather, they set it aside and accessed it when they noticed something on the public display that intrigued them.

Perceptions of privacy and usefulness evolve We found that the balance between privacy and usefulness was always in flux during our deployment. At first, we found that the fact that the public displays showed little information would mitigate privacy concerns. But in fact users were so concerned with the data mining done by the interest sensor that the public display was unimportant. This changed throughout the deployment, though, and as mentioned above some users began to argue for higher-fidelity personal content to be shown on public displays towards the end of the deployment.

Implicit Sensing may be too unobtrusive After initial deployments, we learned that e-mail is clearly something that users value and thus something over which they desire a high amount of control. We originally designed the system with the assumption that all sensors involved should be as perceptually invisible as possible to mitigate their effect upon user work practice. Thus, we initially built the interest sensor to run entirely as a background process with no interface component whatsoever. However, early pilots showed that this approach failed because, as Lederer *et al.* have shown, user demand for course controls and feedback overwhelmed the need for the system to remain completely unobtrusive [36].

Even after integrating these changes into our design, though, we still had difficulty selling the system to many potential users. In particular, these users rejected the very notion of any entity analyzing their e-mail. We attempted to negotiate the amount of information the interest sensors would send to the public displays: from phrases extracted from document contents to only general categories. But with these users

the argument was moot: they simply objected to anything looking at their e-mail regardless of its effects. Interestingly, the terminology used by some users to describe their attitude toward the interest sensor was distinctly human ("I don't want this little guy looking around").

However, as the deployment progressed, interviews revealed that users were increasing less concerned with privacy issues over time. In fact, after the system had been deployed in full for three weeks, users discussed a desire for the public display to show sentences from personal e-mail rather than only words and phrases. Furthermore, some users argued for a lightweight control that would allow them to flag particular phrases for capture when composing an e-mail. Thus, it is important not only to allow users to adapt to implicit sensors over time but to anticipate that they will desire hooks allowing them to control those sensors.

Recruit a Local Champion It is important to identify a person in deployment groups knowledgeable about the group and capable of selling the technology to administrators and directors while eliciting grass-roots support for the adoption of the technology. In our deployment, the only successful installation outside of our lab benefited from such a person. This position is similar to informants used in ethnographies [48], social stars [72] and opinion leaders [60]. However, while informants are primarily important because of the knowledge they have about the group's current status and history, it is more important that the local champion understand how to explain the technology to each individual. This kind of embedded knowledge is difficult to extract through even thorough ethnographies. Also, while a star tends to be the locus of attention for the group, it is more important that the champion not be seen as part of an elite class pushing the technology: the champion should be well embedded in the crowd.

Local champions are vitally important because they can help the system scale by managing local operation of the system. Additionally, they can help to speed up acceptance of a system, leading more quickly to a state in which the system has been effectively integrated into work practice. The expectation of a local champion should be treated as such from the beginning of the experiment design. They should be given more or better benefits for participating in the study and should be given different instructions from the rest of the members.

Practice Participatory Design It is important to integrate deployment groups into the design development process so that they develop an understanding of the goals of the project as well as develop a sense of ownership of the project. We found in our deployments that users struggled to develop a conceptual model of the system. But moreover, early attempts revealed that the users felt that they were involved in an experiment that did not hold any benefits for them: they felt as though they were "guinea pigs" testing an already developed system. A better model is to integrate these users into the design process earlier [69], recruiting them for low fidelity testing and other iterative design steps.

It is difficult to scale intensive participatory designs. But as Dourish notes [21], it is important not only to practice participatory design but also to design applications that are themselves adaptable. It is important to anticipate that users will go through a similar process of realigning themselves with technology in every deployment just as they do during the participatory design. Such flexibility addresses both scale and generalizability. That is, following a participatory design the resultant system may seem as though it solves only a very isolated problem. Thus, it is important to design the system with enough flexibility to allow users to adapt it over time.

Minimize Reinstallations Upon deploying a system several needs and bugs will arise that were not previously anticipated and software and hardware must be fixed in-situ. Certainly steps can be taken to minimize these errors, but some are an inevitable result of testing the system in a different context. However, it is important that such updates are minimized to maintain the unobtrusiveness of the system. One way to mitigate the effect of these reinstallations is to prefer public and remotely accessible (or accessible by the local champion) components to purely private ones. In many cases, such as cellular phone deployments, it is necessary to install private applications, but they should be as lightweight as feasible. Users would much rather be told that there was a change in the system than have to manifest it.

Match Between Questions Asked and Questions Answered When interviewing participants during our deployment of the hebb system, we found that some subjects would answer questions about the system's usefulness quite differently if they viewed the system as a "demo to play with" for a month versus it being a constant, and professionally supported, system. We had to make sure to explore both views with subjects before recording data.

4 Discussion

Many of the lessons learned from evaluating the three case studies overlap. In particular, issues of scale showed up in all three evaluations. Issues relating to acceptability and occurrence of errors and ambiguity showed up in all the evaluations as well. Issues regarding integration into daily life, unobtrusiveness, and implicitness showed up in the second two evaluations. These observations lead us to suggest challenges for Ubicomp evaluation.

Challenge # 1 – Applicability of metrics:

Ubicomp applications may not have the same goals as other technologies and thus may require measurement of different metrics of success. Scholtz and Consolvo have identified several metrics important for Ubicomp evaluation, but also point out that their metrics may vary in appropriateness per the system being evaluated [64]. Analysis of our case studies revealed several important evaluation metrics that we did not originally test for, including comfort, privacy, predictability, accuracy, adoption and adaptability. But the appropriateness of each metric varied across studies. For example, a better *a priori* measurement of the value of adopting the Nutrition Tracking system for a specific user would have led to more useful data. In contrast, PALPlates could have benefited more from measurements of user control and customization. Also, because of the implicit transfer of personal information, privacy was a much more important metric for hebb than for the other studies. Given the large number of metrics one could measure in evaluating a Ubicomp system (Scholtz and Consolvo identified 24 metrics), it would be difficult to measure them all and thus more work needs to be done to understand under what conditions each metric is appropriate.

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. An early Ubicomp system deployed in a classroom, 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 [54] 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 champion can help speed up acceptance and help mitigate time spent on system maintence.

Challenge #3 – Ambiguity:

Ubicomp systems are typically sensing-based systems. As Bellotti *et al.* discuss in their article, "Making Sense of Sensing Systems" [9], 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 [20, 25]. 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. We found that the presence of a local champion can help to mitigate this effect while a system is being iterated on and before it has been perfected.

Challenge #4 – 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, we found it difficult to predict how a system would integrate into daily life in the paper prototype of the nutrition project. 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 [67, 58, 23].

An example of this difficulty can be found in the domain of peripheral display design. As Weiser states in his seminal article on Ubiquitous Computing: "Most of the computers will be invisible in fact as well as metaphor [73]." 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 centers on the idea of invisibility, or readiness-to-hand [76]. As Tolmie [57] and Star [62] 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. While it is not possible for any system to be consistently ready-to-hand, it is important to anticipate and design for breakdowns by exposing state and allowing direct control (*i.e.*, addressing system intelligibility and accountability [9]).

In our studies of hebb, we found that unobtrusiveness could be mitigated by several modifications to our original deployment strategy. In particular, the presence of a local champion, and the use of participatory design, could help to speed up integration of our system into daily life. Additionally, as we evolved the system (using a living laboratory approach), it was necessary to balance evolution against interruptions caused by reinstallations.

4.1 Summary

Given the challenges raised above, we believe that existing evaluation techniques would benefit from modifications before being applied to Ubicomp applications. Note that this does not appear to be an issue for the formative evaluations done before a prototype exists, but rather for the iterative explorations that begin once the first design sketches have been completed.

For very early stage techniques such as heuristic evaluation and paper prototyping, modifications are likely to involve a focus on new metrics [64] and guidelines [41]. As deployment occurs, our experience shows that recruiting a local champion is one strategy for increased success. Another strategy involves the application of participatory design. Lastly, constant adaptation appears to be necessary to deployment success, especially since the difficulty of getting complete feedback before deployment means that systems will not be highly refined when they are deployed.

5 Conclusions

We have shown that evaluation is a problem for Ubiquitous Computing application development, particularly at the early stages of design. Because of a dearth of available techniques, Ubicomp application developers rarely iterate on their designs. From a research perspective, this means that as a community we know less about what makes a Ubicomp application successful than we should. From a development perspective, this makes it harder to develop applications that truly meet the needs of their end users. The contribution of

this paper is a study of issues that arose when evaluating three different Ubicomp systems at different stages of design, and the resulting challenges for Ubicomp evaluation that emerged from those issues.

References

[1] G. 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] G. D. Abowd, C. G. Atkeson, A. F. Bobick, I. A. Essa, B. MacIntyre, E. D. Mynatt, and T. E. Starner. Living Laboratories: The Future Computing Environments Group at the Georgia Institute of Technology. In *Extended Abstracts of the Proceedings of the 2000 Conference on Human Factors in Computing Systems (CHI 2000)*, pp. 215–216, The Hague, Netherlands, April 1-6 2000. ACM Press.

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

[4] G. D. Abowd, E. D. Mynatt, and T. Rodden. The human experience. *Pervasive Computing*, 1(1):48–57, 2002.

[5] 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.

[6] L. Bartram and M. Czerwinsky. Design and evaluation of notification interfaces for ubiquitous computing. Ubicomp 2002 Workshop 9, September 29-October 1 2002.

[7] 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.

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

[9] V. Bellotti, M. Back, W. K. Edwards, R. E. Grinter, D. A. Henderson Jr., and C. V. Lopes. Making sense of sensing systems: five questions for designers and researchers. In L. Terveen, D. Wixon, E. Comstock, and A. Sasse, editors, *Proceedings of the CHI 2002 Conference on Human Factors in Computing Systems (CHI-02)*, pp. 415–422, New York, Apr. 20–25 2002. ACM Press.

[10] V. Bellotti and I. Smith. Informing the design of an information management system with iterative fieldwork. In *Proceedings of DIS*, pp. 227–237, 2000.

[11] S. Carter and J. Mankoff. The design of hebb, a peripheral system supporting awareness and communication, and a study of its impact on small, distributed groups. *In submission*, 2004.

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

[13] C. D. Chandler, G. Lo, and A. K. Sinha. Multimodal theater: Extending low fidelity paper prototyping to multimodal applications. In *Proceedings of ACM CHI'02 Conference on Human Factors in Computing Systems*, Student Posters, pp. 874–875. ACM Press, 2002.

[14] A. Chang, B. Resner, B. Koerner, H. Wang, and H. Ishii. Lumitouch: An emotional communication device. In *Extended Abstracts of Conference on Human Factors in Computing Systems (CHI '01)*, pp. 313–314, Seattle, Washington, March 2001. ACM Press.

[15] C. Chewar and D. S. McCrickard. Adapting UEMs for notification systems. Presented at Design and evaluation of notification interfaces for ubiquitous computing, Ubicomp 2002 workshop 9, September 29-October 1 2002.

[16] S. Consolvo, L. Arnstein, and B. R. Franza. User study techniques in the design and evaluation of a ubicomp environment. In *Proceedings of Ubicomp 2002*, volume 2498 of *Lecture Notes in Computer Science*, pp. 73–90, September 29-October 1 2002.

[17] A. Crabtree, T. Hemmings, and T. Rodden. Finding a place for ubicomp in the home. In *Ubicomp 2003*, p. To Appear, 2003.

[18] 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*, pp. 263–269, Tokyo, Japan, July 2001. IFIP, IOS Press.

[19] N. Dahlback, A. Jonsson, and L. Ahrenberg. Wizard of oz studies – why and how. In *Proceedings of the 1993 International Workshop on Intelligent User Interfaces*, Session 7: Design & Evaluation, pp. 193–200, 1993.

[20] A. Dey, J. Mankoff, G. Abowd, and S. 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)*, pp. 121–130, New York, October 27–30 2002. ACM, ACM Press.

[21] P. Dourish. What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8(1), 2004.

[22] S. Dumais and M. Czerwinsky. Building bridges from theory to practice. In *Proceedings of HCI International 2001, 9th Conference on Human-Computer Interaction*, pp. 1358–1362, New Orleans, LA, USA, August 5-10 2001. Lawrence Erlbaum Associates.

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

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

[25] J. Heer, N. Good, A. Ramirez, M. Davis, and J. Mankoff. Presiding over accidents: System mediation of human action. In *Proceedings of CHI'04, CHI Letters*, volume 6, p. To Appear. ACM, 2004.

[26] J. M. Heiner, S. E. Hudson, and K. Tanaka. The information percolator: Ambient information display in a decorative object. In *ACM Symposium on User Interface Software and Technology*, pp. 141–148. ACM Press, November 1999.

[27] L. E. Holmquist, K. Höök, O. Juhlin, and P. 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.

[28] S. Hudson, J. Fogarty, C. Atkeson, J. Forlizzi, S. Kielser, J. Lee, , and J. Yang. Predicting human interruptibility with sensors: A wizard of oz feasibility study. *Proceedings of the CHI'03 Conference on Human Factors in Computing Systems, CHI Letters*, 5(1):257–264, April 05–10 2003.

[29] H. Hutchinson, W. Mackay, B. Westerlund, B. B. Bederson, A. Druin, C. Plaisant, M. Beaudouin-Lafon, S. Conversy, H. Evans, H. Hansen, N. Roussel, and B. Eiderback. Technology probes: inspiring design for and with families. *Proceedings of the ACM CHI'03 Conference on Human Factors in Computing Systems, CHI Letters*, 5(1), April 05–10 2003.

[30] INSIGHT 19. Beliefs and attitudes of americans toward their diet. USDA Center for Nutrition Policy and Promotion, `http://www.usda.gov/cnpp/insights.htm`, June 2000.

[31] INSIGHT 20. Consumption of food group servings: People's perceptions vs. reality. USDA Center for Nutrition Policy and Promotion, `http://www.usda.gov/cnpp/insights.htm`, October 2000.

[32] S. Intille, C. Kukla, and X. Ma. Eliciting user preferences using image-based experience sampling and reflection. In *Extended Abstracts of the Conference on Human Factors in Computer Systems*, pp. 738–739. ACM Press, 2002.

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

[34] H. Ishii, S. Ren, and P. Frei. Pinwheels: Visualizing information flow in architectural space. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI '01)*, pp. 111–112, Seattle, Washington, March 2001. ACM Press.

[35] H. Ishii, C. Wisneski, S. Brave, A. Dahley, M. Gorbet, B. Ullmer, and P. Yarin. Ambient displays: Turning architectural space into an interface between people and digital information. In *Proceedings of the First International Workshop on Cooperative Buildings (CoBuild '98),*, pp. 22–32. Springer-Verlag, February 1998.

[36] S. Lederer, J. Hong, X. Jiang, A. Dey, J. Landay, and J. Mankoff. Towards everyday privacy for ubiquitous computing. *Technical Report UCB-CSD-03-1283, Computer Science Division, University of California, Berkeley*, 2003.

[37] L. Liu and P. Khooshabeh. Paper or interactive? a study of prototyping techniques for ubiquitous computing environments. In *Proceedings of CHI 2003*, pp. 1030–1031. ACM, 2003.

[38] E. H. 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 '02)*. ACM Press, May 2002.

[39] P. P. Maglio and C. S. Campbell. Tradeoffs in displaying peripheral information. In *Proceedings of ACM CHI 2000 Conference on Human Factors in Computing Systems*, pp. 241–248, 2000. 917 KB.

[40] L. Mamykina, E. Mynatt, and M. A. Terry. Time aura: Interfaces for pacing. In *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, Visions of Work, pp. 144–151, 2001.

[41] J. Mankoff, A. K. Dey, G. Hsieh, J. Kientz, M. Ames, and S. Lederer. Heuristic evaluation of ambient displays. *Proceedings of the ACM CHI'03 Conference on Human Factors in Computing Systems, CHI Letters*, 5(1):169–176, April 05–10 2003.

[42] J. Mankoff, G. Hsieh, H. C. Hung, S. Lee, and E. 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.

[43] J. Mankoff and B. Schilit. Supporting knowledge workers beyond the desktop with PALPlates. In *Proceedings of CHI'97*, pp. 550–551. ACM, 1997.

[44] D. S. McCrickard, R. Catrambone, C. M. Chewar, and J. 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.

[45] D. S. McCrickard, R. Catrambone, and J. T. Stasko. Evaluating animation in the periphery as a mechanism for maintaining awareness. In M. Hirose, editor, *Proceedings of Interact'01: Human-Computer Interaction*, pp. 148–156, Tokyo, Japan, July 2001. IFIP, IOS Press.

[46] D. S. McCrickard and C. M. Chewar. Attuning notification design to user goals and attention costs. *Communications of the ACM*, 46(3), March 2003.

[47] D. McGee, P. R. Cohen, R. M. Wesson, and S. Horman. Comparing paper and tangible, multimodal tools. In L. Terveen, D. Wixon, E. Comstock, and A. Sasse, editors, *Proceedings of the CHI 2002 Conference on Human Factors in Computing Systems (CHI-02)*, pp. 407–414, New York, Apr. 20–25 2002. ACM Press.

[48] D. R. Millen. Rapid ethnography: Time deepening strategies for hci field research. In *Proceedings of DIS*, pp. 280–286, 2000.

[49] A. H. Mokdad, M. K. Serdula, W. H. Dietz, B. A. Bowman, J. S. Marks, and J. P. Koplan. The spread of the obesity epidemic in the united states, 1991-1998. *Journal of the American Medical Association*, 282(16):1519–1522, October 1999.

[50] T. P. Moran, W. van Melle, and P. Chiu. Tailorable domain objects as meeting tools for an electronic whiteboard. In *Proceedings of ACM CSCW'98 Conference on Computer-Supported Cooperative Work*, From Single-Display Groupware to Mobility, pp. 295–304, 1998.

[51] M. J. Muller. Retrospective on a year of participatory design using the PICTIVE technique. In *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, Participatory Design, pp. 455–462, 1992.

[52] E. Mynatt, M. Back, R. Want, M. Baer, and J. B. Ellis. Designing Audio Aura. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '98)*. ACM Press, 1998.

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

[54] J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. In *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, Methodology, pp. 249–256, 1990.

[55] S. Oviatt. Multimodal interfaces for dynamic interactive maps. In *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, volume 1 of *PAPERS: Multi-Modal Applications*, pp. 95–102, 1996.

[56] S. Oviatt. Multimodal system processing in mobile environments. In *Proceedings of the ACM UIST 2000 Symposium on User Interface Software and Technology*, pp. 21–30. ACM Press, November 2000.

[57] e. a. P. Tolmie, J. Pycock. Unremarkable computing. In *Proceedings of CHI*, pp. 399–406, 2002.

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

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

[60] E. Rogers. *Diffusion of Innovations*. Free Press, 2003.

[61] 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.

[62] K. R. S. L. Star. Steps towards an ecology of infrastructure: complex problems in design and access for large-scale collaborative systems. In *Proceedings of CSCW*, pp. 253–264, 1994.

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

[64] J. Scholtz and S. Consolvo. Towards a discipline for evaluating ubiquitous computing applications. *Technical Report IRS-TR-04-004*, 2004.

[65] J. Scholtz, E. Tabassi, S. Consolvo, and B. Schilit. User centered evaluations for ubiquitous computing systems: Best known methods. Ubicomp 2002 Workshop 2, September 29-October 1 2002.

[66] A. K. Sinha and J. A. Landay. Embarking on multimodal interface design. In *IEEE International Conference on Multimodal Interfaces, Poster*, Pittsburgh, PA, October 2002, 2002.

[67] M. Steves, E. Morse, C. Gutwin, and S. Greenberg. A comparison of usage evaluation and inspection methods for assessing groupware usability. In *Group'01*, pp. 125–134, Boulder, CA, September 2001. ACM Press.

[68] J. Trevor, D. M. Hilbert, and B. N. Schilit. Issues in personalizing shared ubiquitous devices. In *Proceedings of Ubicomp 2002*, volume 2498 of *Lecture Notes in Computer Science*, pp. 56–72, 2002.

[69] T. M. e. a. V. Anantraman. Handheld computer for rural healthcare, experiences in a large scale implementation. In *Proceedings of development by design*, pp. 1–10, 2002.

[70] 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.

[71] K. Vredenburg, J.-Y. Mao, P. W. Smith, and T. Carey. A survey of user-centered design practice. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 471–478. ACM Press, 2002.

[72] S. Wasserman and K. Faust. *Social network analysis: methods and applications*. Cambridge University Press, 1994.

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

[74] T. White and D. Small. An interactive poetic garden. In *Extended Abstracts of Conference on Human Factors in Computing Systems (CHI '98)*, pp. 335–336. ACM Press, 1998.

[75] C. Wickens and J. Hollands. *Engineering psychology and human performance*. Prentice Hall, Upper Saddle River, NJ, USA, 2000.

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

[77] L. A. T. Xiadong Jiang, Jason I. Hong and J. A. Landay. Ubiquitous computing for firefighters: Field studies and prototypes of large displays for incident command. In *Proceedings of CHI 2004, CHI Letters*, volume 6, p. To Appear. ACM, 2004.