GVU Technical Report GIT-GVU-98-01

Ubiquitous Computing: Defining an HCI Research Agenda for an Emerging Interaction Paradigm

D. Salber, A. K. Dey and G. D. Abowd
GVU Center & College of Computing
Georgia Institute of Technology, Atlanta, GA, USA
Phone: +1 404 894 7512, Fax: +1 404 894 2970
Email: {salber, anind, abowd}@cc.gatech.edu

Abstract

Ubiquitous computing (ubicomp) is an emerging paradigm for interaction between people and computers. A guiding principle of ubicomp is to break away from desktop computing to provide computational services to a user when and where required. Although there has been a lot of experimental work in ubicomp, there has been little effort to define an agenda in ubicomp for HCI researchers. In this paper, we attempt to remedy that problem by defining the space of ubicomp applications in terms of the level of user mobility and transparency of interaction. Increases in user mobility will come with technological advances, but increased interaction transparency will come only with breakthroughs in HCI research. We conclude the paper with a discussion of two functional themes that we have found important across a number of ubicomp systems —context-awareness and automated capture, integration and access. Each of these themes raises special HCI issues and, together with the taxonomy for ubicomp applications, defines a clearer agenda for HCI research in ubiquitous computing.

Keywords

Ubiquitous computing, taxonomy, human-computer interaction, context-aware computing, capture, integration, and access.

1 INTRODUCTION

The interest in ubiquitous computing (or ubicomp, for short) has surged over the past few years, thanks to some influential writings and plenty of experimental work. The history of computing is filled with examples of radical paradigm shifts in the way humans interact with and perceive technology. The vision of ubiquitous computing —first expressed by Weiser (Weiser, 1991) and grounded in experimental work done at Xerox PARC— holds the promise of yet another interaction paradigm shift.

Many researchers have intuitive and rather informal definitions of ubiquitous computing. One definition that we prefer is that ubicomp is an attempt to break away from the current paradigm of desktop computing to provide computational services to a user when and where required. Rather than force the user to search out and find the computer's interface, ubiquitous computing suggests that the interface itself can take on the responsibility of locating and serving the user. But this informal definition needs refinement in order to distinguish ubicomp from related areas of augmented reality, wearable computing and mobile computing. One purpose of this paper is to provide a definition of what is, and more importantly what is not, within the domain of ubiquitous computing. We do this by examining two dimensions of ubiquitous computing first suggested by Weiser— mobility and transparency.

Much of the research in ubiquitous computing has focused on technological infrastructure and articles that outline research issues focus on systems and networking, and a little bit on social concerns (Weiser, 1993) (Spreitzer et al., 1993). There are many issues of concern to the HCI researcher, and it is another goal of this paper to outline those research issues. First, for those interested in building ubicomp applications, we have identified two common functional services that are shared by many different applications: context-awareness; and automated capture, integration and access of live experiences. These functional services are necessary, but currently missing, software infrastructure that would bootstrap widespread ubicomp system development. Second, for those interested in understanding the impact of ubicomp on our everyday lives, there are research issues related to these two functional services that require serious consideration. Some of these research issues can be understood within the mobility and transparency dimensions.

Three ubiquitous computing applications

As Weiser points out, "Applications are of course the whole point of ubiquitous computing." (Weiser, 1993) Fueled by this statement, three years ago we initiated a group at Georgia Tech, the Future Computing Environments (FCE) Group, to investigate and invent applications of ubiquitous computing technology. Here we will briefly describe three applications that we have developed. These projects will be referred to throughout this paper. It is important to understand that most of our opinions on ubicomp have been formulated based on much experience designing, implementing, *and using* a number of applications. We take time now to briefly describe three of our projects.

Classroom 2000

The Classroom 2000 project is investigating the impact of ubiquitous computing on university education (Abowd et al., 1998) (Brotherton et al., 1998). We have instrumented a single classroom with a large-scale, pen-based electronic whiteboard that enables an instructor to present and annotate a standard lecture, using a blank surface, a prepared presentation or a series of Web pages as the background. Our software captures much of the lecturer's activity and timestamps it. In addition, the room is equipped with digital recording infrastructure, and we automatically generate Web-accessible notes that coordinate the captured lecture notes with the audio/video recordings and the Web URLs accessed by the lecturer during the class. The result is an environment that attempts to relieve students of the burden of copious, and often incomplete, note-taking so that they can engage more directly in the classroom experience.

Cyberguide

Cyberguide is a handheld, mobile tour guide that assists users in visiting various parts of the Georgia Tech campus (Long et al., 1996) (Long et al., 1996) (Abowd et al., 1997). We have produced a number of prototypes that use either indoor or outdoor positioning data to inform the system where the user is located. Cyberguide uses this information to provide more salient information to the visitor about the surrounding space, such as building names for a campus tour or information about exhibits for visitors to our regular research open houses. Some versions produced a Web-based summary of a visitor's tour that would remind them of the various sites that were visited and preserved comments that they made along the way. We also built a prototype that kept track of the location of multiple visitors and displayed that information upon request on top of a single user's electronic map (Pinkerton, 1997).

Domisilica

The Domisilica project is concerned with augmenting a home in order to provide automation of mundane tasks (e.g., turning down the stereo when the phone rings). This constitutes the core of the Domisilica system. Extensions provide remote access to information (Mankoff et al., 1997) (Mankoff et al., 1998). We have built systems that allow physical activity in the home, such as stocking the refrigerator, to automatically update information in a virtual environment. We have also built a Web-based graphical interface to the virtual environment that allows for remote interaction with the physical environment as well as with other virtual guests to the home.

Overview of paper

In the next section of this paper, we will first define the boundary for ubicomp based on the degree of user mobility and the degree of interaction transparency provided by the system. The resulting framework allows us to elicit relationships between ubicomp and related emerging research fields such as mobile computing or smart environments. In section 3, we examine further the space of ubicomp systems along the user mobility and interaction transparency dimensions. We identify the position of the state of the art of ubicomp in terms of these two dimensions and suggest directions for further ubicomp research. From our experience, we uncover two major classes of generic functional services relevant to ubicomp in section 4. Using the mobility and transparency dimensions of our framework to drive our analysis, we point out HCI issues relevant to these functional services.

2 DEFINING UBIQUITOUS COMPUTING

We will begin by trying to define the salient features of a ubiquitous computing system and clarify the relationship between this area and other emerging research fields, such as mobile computing, wearable computing, augmented reality and smart (or "intelligent") environments. Once we have established a clear boundary between what is and is not a ubiquitous computing system, we will, in the next section, further refine the space of ubicomp applications.

A ubiquitous computing system consists of (a) a (possibly heterogeneous) set of computing devices; (b) a set of supported tasks; and (c) some optional infrastructure (e.g., network, GPS location service) the devices may rely on to carry out the supported tasks.

Unlike traditional desktop applications with a graphical user interface, ubicomp applications force us to take a rather general view of a system. In traditional GUIs, the interactive system is the desktop computer and a fixed set of input/output devices. The emphasis is on combining software components to provide services to the user. With ubicomp systems, we are concerned not only with software services but also with devices and how to combine them.

An ontological framework

What are the inherent features of a system that make it a ubiquitous computing system? According to (Weiser, 1991), ubicomp is characterized by two main attributes:

- *ubiquity:* interaction with the system is available wherever the user needs it;
- *transparency:* the system is non-intrusive and is integrated into the everyday environment.

In Weiser's view, ubiquity denotes the universal availability of computation throughout multiple ubicomp systems in the user's environment. Since we look at a single ubicomp system at a time, we're rather concerned with the mobility allowed to the user by the system. We refine Weiser's fairly intuitive attributes into two dimensions —user mobility and interaction transparency— that will provide a clear boundary for ubiquitous computing and express the relationship between ubicomp and other emerging areas.

User mobility

A system allows interaction with the user within a given range of possible locations. We define the *user mobility* dimension of our framework to reflect the freedom the user has to move about when interacting with the system. Desktop computing allows no user mobility; she has to sit and stay in front of the fixed machine. Some systems allow more freedom of movement to the user when interacting with the system. Systems relying on an instrumented environment like the Active Badge system (Want et al., 1992), Classroom 2000, or Domisilica, function wherever the user is in the space covered by the system (e.g., a building, a room, a house). When using a system based on a standalone portable device, user mobility is unconstrained. When the portable device relies on an infrastructure to provide services, the mobility of the user is usually constrained to the coverage zone of the infrastructure. For example, systems relying on GPS service constrain mobility because GPS is not available indoors or may be obscured by buildings or tunnels.

Interaction transparency

In the ubicomp literature, transparency is used interchangeably with the terms invisibility, embodiment in the environment, intuitiveness, anticipation of the user's intent, affordance, and peripheral awareness. Interaction transparency applies to the system's interface and reflects the conscious efforts and attention the system requires of the user, either for operating it or for perceiving its output.

Most interfaces today lack interaction transparency. To perform a task with the system, the user must consciously perceive, understand and manipulate an interface which is conceptually separate from the task being performed. In the terms of (Hutchins et al., 1985), there is still an important articulatory distance for execution and evaluation, although direct manipulation alleviated some of it. Pushing buttons using a mouse, moving windows on the screen offer poor directness compared to their physical analogies of pushing a real button or shuffling through pieces of paper. The graphical user interface remains in the focus of the user throughout the interaction.

A transparent interface on the other hand, disappears from the user's focus so she can concentrate on the actual task at hand. Offering greater manipulation directness is a way to achieve some level of transparency and can be found in all ubicomp systems. PARCTabs units feature actual buttons, the Classroom 2000 lecturer writes using an electronic pen on a Liveboard which is physically very close to an actual whiteboard. Another way to achieve interaction transparency is to relieve the user of some tasks by providing task migration (Dix et al., 1998). When a task is migrated from the user to the system, the system takes responsibility for performing the task on behalf of the user. For example, a user request to print a document can leave it to the system to figure out the most appropriate printer to use, based on the user's current location. In the Audio Aura system (Mynatt et al., 1997), information relevant to the user is provided without the user having to explicitly request it. In Classroom 2000, the beginning of a lecture requires a synchronization of all recording devices. Transparency in this case means that the task of synchronization should not be a concern of the user but should be handled appropriately by the system. In general, migration of administrative tasks from user to system provides a more transparent interface.

Transparency deals with output as well. When the system presents information to the user, it may provide output without requesting the full attention of the user. Audio cues in Audio Aura for example, provide information to the user in a non-intrusive way; the user is free to attend to them or to ignore them. In the Portholes video communication system (Dourish et al., 1992), a video mosaic provides the user with awareness of her colleagues' presence and activities. Here again, Portholes sits in the background and does not request the user's attention. If the user wishes to establish communication with a colleague, however, she can focus on the information provided by the Portholes window. In these cases, the interface provides peripheral awareness of output, and the user can summon that information to their attention on request.

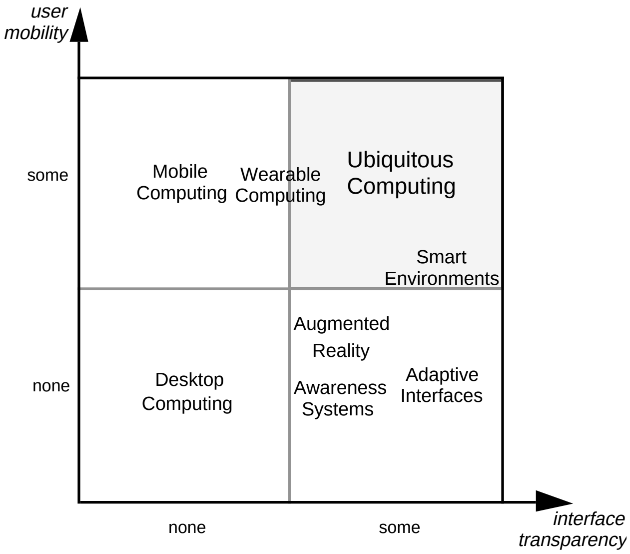

Figure 1 The mobility/transparency matrix.

The boundary for ubiquitous computing

Figure 1 presents our ontological framework. It consists of the two dimensions we just defined: user mobility and interface transparency. The values for each dimension are simply "none" and "some". A ubiquitous computing system is one that provides some user mobility and some interaction transparency.

We can contrast that with desktop computing, which offers no mobility and, for the most part, no transparency to the end user.

With this framework, we can characterize other research streams relevant to HCI and ubiquitous computing. Adaptive interfaces for example, aim at anticipating the user's actions (Cypher, 1993) (Schneider-Hufschmidt et al., 1993), but they are usually concerned with enhancing a desktop interface. Thus they offer more transparency than traditional GUIs but don't allow for user mobility. Similarly, awareness systems like Portholes (Dourish et al., 1992), provide information in a transparent way but are not designed to accommodate mobile users. Although Portholes is a communication system available throughout a building, the system is designed to be used while sitting at one's workstation. Augmented reality systems, by adapting the information presented to the user's position in a given environment (e.g., when attending a photocopier in the KARMA system (Feiner et al., 1993)) also provide transparency. The mobility they allow to the user however is very limited: current designs don't allow for moving outside the scope of some object of interest. Current smart environments instrument rooms to capture the user's actions and react accordingly. They also provide some level of transparency and limited mobility (sometimes up to a room but more often a dedicated area in front of sensors) (Coen, 1998) (Cooperstock et al., 1995).

Current mobile computing research aims at providing desktop-like systems to the user, wherever she may be. Such systems don't aim at providing more transparent interfaces than desktop computing. A related area of research, wearable computing, relies on smaller devices that the user can actually wear. Due to the inherent constraints of the interaction (limited input devices, usage conditions not appropriate for complex input), wearable systems tend to provide more directly manipulable interfaces or relieve the user from some tasks. Thus, wearable computing gradually progresses towards more transparent interfaces. In our view, systems which stand across two different fields, one pushing mobility and the other transparency are actually ubicomp systems. For instance, the Boeing airplane maintenance system (Esposito, 1997) which combines wearable computing and augmented reality falls in the ubicomp category.

3 CHARACTERIZING UBIQUITOUS COMPUTING

The mobility/transparency matrix of Figure 1 clearly defines the boundary for ubiquitous computing. We will now examine more closely the classification of systems that fall in the upper right quadrant of Figure 1. With this framework we provide an overview of the state of the art in ubicomp and identify further research directions.

User mobility revisited

On the user mobility axis, we now consider two values: constrained mobility and full mobility. *Constrained mobility* allows movement in a well-defined and limited space. Systems that constrain the mobility of the user typically have a coverage zone outside of which interaction is not possible. A cellular phone and the associated network are an example of a system that constrains the user's mobility (Note that the cellular phone by itself may be operable outside the coverage zone, e.g., to add an entry to the phonebook). Systems in this category are not necessarily carried by the user. The capture system in Classroom 2000, for example, allows users to freely move about in the classroom. Similarly, an indoor prototype of Cyberguide provides service only in a limited coverage zone. Instrumenting a closed space like a building or a conference room with sensors and a communication infrastructure has been a popular approach so far in ubicomp.

In Domisilica, the whole house is instrumented and provides services in whatever room the user may be. However, some information in Domisilica, like the contents of the display on the refrigerator door, is available to authorized users anywhere on the Internet. We note, however, that we have to consider a different system than the core in-house Domisilica in this case and include a remote terminal acting as a Web client, and a network infrastructure between the terminal to the house. An interesting evolution would be the use of a handheld device with truly global communication capabilities that would allow access to the refrigerator door display from any location in the world, allowing a parent to leave messages for the rest of the family while on a far-away business trip. Such a system would be classified as full mobility.

Full mobility defines systems that pose absolutely no constraints on the geographical location of the user. Currently, this category of systems is mainly restricted to standalone systems that don't rely on the availability of some additional infrastructure. An example would be PDAs that don't rely on a network infrastructure for most tasks. Actually, most "global" infrastructures available today constrain the user's mobility in some way: the GPS system for example doesn't operate indoors. It is not clear if low-earth orbit (LEO) satellite-based communications will be available indoors as well. Alternatively, hybrid systems which could adapt to LEO-based networks as well as indoor networks may provide a step towards truly full mobility. Nevertheless, "dark spots" in the coverage zone, like underground urban areas (subway, parking lots, tunnels) may remain uncovered.

It is interesting that many real-world objects that we consider ubiquitous technology fall into the "full mobility" category. Pen and paper, books, wristwatches, or glasses for example can be used anywhere. However, current technology limits user mobility. Technology advances and market demand due to compelling applications may alleviate this limitation in the future.

We have identified two degrees of mobility a ubicomp system allows the user: constrained or full. We now turn to the second dimension of the taxonomy: transparency.

Interface transparency revisited

To refine the concept of transparency, we propose two values along the transparency dimension of our taxonomy: syntactic and semantic.

Syntactic transparency relieves the user of syntactic tasks, that is, tasks that are introduced by the system itself. Typical syntactical tasks are: saving a file, retrieving email from a server, or in a GUI, moving or scrolling windows. When performing a syntactical task, the user doesn't actually "do some work" but merely wrestles with the system's specifics (its syntax) to be able to later perform some "real" semantic task (e.g., read and reply to email, type in a previously obscured window, etc.). For example when jotting notes on a Newton PDA, the user doesn't have to worry about saving: the memory-based device alleviates this syntactical operation. In the real world, the point-and-shoot camera relieves the user of setting the snapshot parameters. In the ubicomp world, Classroom 2000 provides an example of syntactic transparency by relieving the user of administrative tasks associated with initialization and synchronization. When the lecturer is ready to begin the class, a simple push on a button allows her to start all the logging streams simultaneously. One service implemented with PARCTabs or Active Badges is that the system automatically directs print requests to the printer nearest to the user, relieving the user of choosing it explicitly. In summary, for a system to be syntactically transparent, it must relieve the user of tasks that are related to the intricacies of its operation.

Syntactic transparency also applies to output. When syntactically transparent, a system makes the user aware of its workings in a non-intrusive way. For example, the green light indicating "on" on a kitchen appliance doesn't request our attention but we can focus on it to check the system actually works. Weiser's example of the familiar hum of the car engine doesn't request our attention either: we're peripherally aware of it and it's not intrusive except when it's unusual. In ubicomp systems, Weiser's Dangling String (Weiser et al., 1997) stirs according to network traffic. From this syntactic output, users can then infer higher-level information (e.g., many people are working late tonight). Similarly in the outdoor Cyberguide, a discreet on-screen level indicator could indicate the accuracy of the GPS positioning system information. The user could refer to it to check the correct functioning of the system.

Semantic transparency characterizes a system that anticipates the user's intent and performs the task for her. A common real-world example is the automatic sliding door. The system opens the door for the approaching user. A light coupled to a motion sensor that turns on when someone enters a room is also an example of semantic transparency. Classroom 2000 could use environment information to detect the beginning of a lecture and start recording automatically (e.g., when all students turn silent and face the lecturer, or when the lecturer grabs a pen). In the home context, our Domisilica system controls home appliances (VCR, TV, lights) according to the inhabitants context (motion, incoming events like phone calls, etc.)

Semantic transparency for output happens when the system communicates real-world information (as opposed to information about the system itself) in a non-attention-grabbing way. For example, in the

Portholes system, the user is made peripherally aware of the whereabouts of her colleagues. When she needs to contact somebody, she can switch her focus to that information. Audio Aura provides information (e.g., email received, new library acquisitions) depending on the location of the user. Instead of grabbing the user's attention (e.g., using a video screen in the library), it delivers this information using background environmental sounds.

Looking at the current state-of-the-art in ubicomp, we can make two observations: first, full mobility is not attained in current ubicomp systems. The only example we came across that exhibits full mobility is Apple's envisionment of translating glasses (Apple, 1987). Technological constraints in e.g., network infrastructures are certainly an issue here, but some solutions are worth exploring. Hybrid systems that can be used either indoors or outdoors, or systems that provide graceful degradation when the needed infrastructure is no more available are research directions that need to be pursued. Second, transparency is most commonly found at the syntactic level. Even though, it is usually limited to simple uses like location-awareness. Progressing towards greater transparency is a serious challenge for HCI research. In the next section, we identify two generic functional services for ubicomp and outline the HCI issues they raise with regard to mobility and transparency.

4 FUNCTIONAL SERVICES FOR UBIQUITOUS COMPUTING

The dimensions of mobility and transparency have helped us clearly delineate the boundaries of ubiquitous computing research and identify some of the difficult HCI challenges for improving the quality of ubicomp applications. Since much of our understanding of ubicomp has come through the experience of building and using applications, we see the need to develop more general infrastructures to facilitate the rapid development of applications. This infrastructure can be in the form of improved hardware technology or software solutions. Since software solutions are more readily reusable than hardware installations, we focus on two general software services that should be the focus of researchers wanting to provide reusable solutions for ubicomp application development. These two services are: context-awareness; and automated capture, integration, and access. We will define each service below and justify their importance through references to our work and the work of others. We will then discuss issues within each theme that directly relate to the dimensions of mobility and transparency and other HCI concerns.

Context-awareness

Definition

Future computing environments promise to free the user from the constraints of stationary desktop computing. Increased user mobility, a defining dimension of ubiquitous computing, suggests that applications should adapt themselves based on knowledge of location. This location can be position and orientation of a single person, many people, or even of a certain set of devices. Location is a simple example of context, that is, information about people or devices that can be used to modify the way a system provides its services to the user community. Location is an example of physical context. Other categories of context include informational (what data is the user focused on), emotional (how a user feels), intentional (what does the user want to do) and historical (what is the record of context over time). Context-aware computing aims to provide maximal flexibility of a computational service based on real-time sensing of any of these forms of context.

Context-awareness is not unique to ubiquitous computing. For example, explicit user models used to predict the level of user expertise are a good example of context-awareness and has been used in many desktop systems. However, context-awareness is a critical feature for supporting interaction transparency of a ubiquitous computing system. Whereas it is a nice feature for desktop-bound applications, greater dynamicity of the context makes context-awareness more of a necessity for mobility-enhanced applications that aim to provide syntactic or semantic transparency. Context-awareness is a key feature in a shift away from "personalized computing", in which users own devices that are tailored to their needs, toward "personalizable computing", in which users need not possess a device in order for it to be tailored to their needs.

Examples

There are many examples of location-aware computing. Besides our own Cyberguide work, there was seminal work in this area done at Olivetti Research Labs, developers of the Active Badge location system (Want et al., 1992), and at PARC, through the PARCTab system (Want et al., 1995) and other location-aware services. A more general programming framework for describing location-aware objects was the subject of Schilit's thesis (Schilit, 1995).

We have investigated an application of informational context used to automatically integrate the behavior of network-based personal information management services (the CyberDesk project (Dey et al., 1998)). We are now looking at extending the context inferencing engine to support applications in which knowledge of the people and place causes automatic modification of services displayed on mobile displays (Dey, 1998). Similar work using informational context to integrate desktop applications has been reported by Apple (Data Detectors (Apple, 1997)) and Intel (Pandit and Kalbag's Selection Recognition Agent (Pandit et al., 1997)).

Advances in computational perception (Essa et al., 1995) and affective computing (Picard, 1995) are making it possible for us to consider the possibility of enabling emotional and intentional context either by instrumenting the environment to perceive information about its occupants or by attaching wearable sensors to the users themselves.

Relating to Mobility and Transparency

As ubicomp applications move toward full mobility, the need for context-awareness increases. Location-awareness is perhaps the simplest form of context-awareness. As we learned from our experience with Cyberguide (Abowd et al., 1997), better context awareness is not always served best by higher precision location services. Determining the focus of attention of a visitor is most important and can be gleaned from rough position and orientation combined with gaze, speech and gesture.

Context-awareness is critical to achieving any level of interaction transparency. In the CyberDesk project, simple recursive translations of a selected string of text to other data types (names, places, URLs) allowed for syntactic transparency. The system produced dynamic buttons that could be used to more quickly invoke various network-based services that would use the selected text as input. Further contextual information, such as the history of interactions by a single user, could then be used to provide semantic transparency that would understand exactly which operation the user intends to invoke and automatically invoke it. For CyberDesk, syntactic transparency rapidly fills the user's screen with relevant actions that could be performed, and semantic transparency aims to pare down those actions to a shorter list of salient actions.

As another example, an intelligent display on the refrigerator door in Domisilica could adapt what is shown based on knowledge of who is looking at the display. Depending on the time of day, it could post different information. Context-awareness can also be used to determine the level of distraction for peripheral awareness services. As impending deadlines approach, discreet reminders can be made more prominent. Another important transparency issue with context-awareness is the ability to determine context by observing natural actions of the user. Work in computational perception and affective computing is aimed at context determination with minimal requirement for explicit user action.

Capture, Integration and Access

Definition

Much of our life is spent listening to and recording, more or less accurately, the many events that surround us, and then trying to remember the important details of one specific event that eludes us. There is a value to using computational resources to augment the inefficiency of human record-taking, especially when there are multiple streams of related information which are virtually impossible to capture as a whole. Computational support can also automate explicit and implicit links between related but separately generated streams of information. Finally, a rich record of a group interaction can support later access to aid in recalling the meaning or significance of past events. Together, automated capture, integration and access tools can remove the burden of doing something we are not good at (recording) so that we can focus attention on things we are good at (indicating relationships, summarizing, and interpreting).

Examples

The Classroom 2000 project is mainly concerned with capture, integration and access in support of lecture-based education. The many streams of activity in a typical lecture -what is being said, what is seen, what is written down on a whiteboard and what is shown on public displays- are combined to provide a rich interactive experience that is becoming increasingly more difficult to capture using traditional pen and paper notes. Some of the Cyberguide prototypes created summaries of when and where a visitor traveled on campus and preserved images taken with a camera or comments made by the visitor that were attached to this temporally- and spatially-indexed travel diary.

Other research teams have used this same notion of capture, integration, and access to facilitate collaborative or personal experiences. Work at Xerox PARC focused on capturing technical meetings to support summarization by a single scribe who was often not well-versed in the subject of the meetings (Minneman et al., 1995) (Moran et al., 1997). More work at PARC (the Marquee system (Weber et al., 1994), together with work at Hewlett-Packard (the Filochat system (Whittaker et al., 1994), Apple (Degen et al., 1992)), and MIT's Media Lab (Stifelman, 1996) demonstrates the utility of personal note-taking with automatic audio enhancement for later review.

Relating to Mobility and Transparency

Our everyday experiences are not confined to fixed locations, so these capture, integration, and access capabilities must be available over a large area. In Classroom 2000 today, constrained mobility is sufficient as the lectures occur within the confined space of a classroom. Similarly, in Domisilica constrained mobility is sufficient within the confined (albeit larger) area of a home. In each of these cases, access to any captured experience occurs outside the confined space, and for that we rely on the Web. Cyberguide, on the other hand, is an application that will require full mobility.

Transparency is a very important consideration, with both positive and negative ramifications. On the positive side, we want to push for maximal, or semantic, transparency. Teachers are not usually motivated enough to spend extra time before, during and after a lecture interacting with complex equipment to record their lecture for the benefit of students in Classroom 2000. Capturing the lecture needs to be as simple as picking up a pen and beginning to talk. The burden should be on the system, not the user, to encode the natural activities of a collaborative experience, so that it can be properly indexed to facilitate later review. We have spent a lot of energy to increase the syntactic transparency of the system. For example, simply knowing the class schedule in a room removes a lot of initialization tasks that were required of the lecturer. The teacher simply opens up the electronic whiteboard application, types in a title for the lecture, authenticates herself with a username and password and clicks on the "Begin lecture" button. At the end of class, closing the application will automatically set in motion the post-production process that creates the audio/video-enhanced Web notes without any further interaction from the teacher. We have not, however, provided for any level of semantic transparency in the system. We would like to detect some higher-level structure in a lecture to facilitate integration and access. For example, a better integration scheme would use more than timing information, linking what is written with the best place in the audio stream that is related to the writing. This is the target of future research.

On the negative side, too much transparency can be problematic, and users must be made aware of what is captured, who will have access to it and why. In addition, there should be ways for the users to control what is being captured and remove segments that they would rather not save. Knowledge of what is being captured can also help the users to adapt their own note-taking behaviors and allow them to be more effective with their time.

5 CONCLUSION

In this paper, we have attempted to define the space of ubiquitous computing applications, using the dimensions of user mobility and interaction transparency. This ontological framework clarifies the relationship between ubiquitous computing and other emerging research areas of interest to HCI researchers. Based on our experience building and using a number of ubicomp applications, we have identified two general functional services ---context-awareness and capture, integration and access for live experiences--- that are relevant across a variety of applications. Each of these services raises issues with respect to user mobility and interaction transparency that can lead HCI research agendas for ubiquitous computing.

From an HCI perspective, the dimension of interaction transparency is most important. Increased user mobility will largely come from improvements in hardware infrastructure, but greater transparency will only come with software solutions partnered with good HCI design practices. The suggestion of general functional services is intended to inspire software researchers to build toolkits to facilitate rapid development of ubicomp applications. Only when we are able to easily deploy large-scale ubicomp systems will we be able to understand how this emerging interaction paradigm shift impacts the relationship between user and system.

6 ACKNOWLEDGMENTS

This work was partially supported by INRIA, NSF, DARPA and numerous industrial sponsors (Motorola, Sun, HP, BellSouth, CNRI, Intel). We wish to thank the numerous FCE faculty and students who took part in the FCE projects. Special thanks are due to Jen Mankoff for constructive comments on previous versions of this paper.

7 REFERENCES

Abowd, G. D., Atkeson, C. G., Brotherton, J., Enqvist, T., Gulley, P. and LeMon, J. (1998) Investigating the capture, integration and access problem of ubiquitous computing in an educational setting, in *Proceedings of the 1998 ACM Conference on Human Factors in Computing Systems*, To appear.

Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R. and Pinkerton, M. (1997) Cyberguide: A Mobile Context-Aware Tour Guide. *ACM Wireless Networks,* **3**, To appear.

Apple (1987) The Knowledge Navigator (video).

Apple (1997) Apple Data Detectors homepage, Apple Research Laboratories.

Brotherton, J. A. and Abowd, G. D. (1998) Rooms Take Note! Room Takes Notes!, in *Proceedings of the 1998 Spring AAAI Symposium on Intelligent Environments*.

Coen, M. (1998) Proceedings of the 1998 Spring AAAI Symposium on Intelligent Environments.

Cooperstock, J. R., Tanikoshi, K., Beirne, G., Narine, T. and Buxton, W. (1995) Evolution of a Reactive Environment, in *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*.

Cypher, A. (1993) Eager: programming repetitive tasks by demonstration, in *Watch what I do: programming by demonstration*, 205-217, MIT Press, Cambridge MA.

Degen, L., Mander, R. and Salomon, G. (1992) Working with Audio: Integrating Personal Tape Recorders and Desktop Computers, in *Proceedings of the ACM 1992 Conference on Human Factors in Computing Systems*, 413-418.

Dey, A. (1998) Context-Aware Computing: The CyberDesk Project, in *Proceedings of the 1998 Spring AAAI Symposium on Intelligent Environments*.

Dey, A., Abowd, G. D. and Wood, A. (1998) CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services, in *Proceedings of the 1998 Intelligent User Interfaces Conference*, 48-54.

Dix, A., Finlay, J., Abowd, G. and Beale, R. (1998) Human-Computer Interaction (2nd edition). Prentice Hall, London.

Dourish, P. and Bly, S. (1992) Portholes: Supporting Awareness in a Distributed Work Group, in *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 541-547.

Esposito, C. (1997) Wearable Computers: Field-Test Observations and System Design Guidelines. *Personal Technologies*, **1**.

Essa, I. and Pentland, A. (1995) Facial Expression Recognition using a Dynamic Model and Motion Energy, in *Proceedings of the International Conference on Computer Vision*, 360-367, IEEE Computer Society, Cambridge, MA.

Feiner, S., MacIntyre, B. and Seligmann, D. (1993) Knowledge-based augmented reality. *Communications of the ACM,* **36(7)**, 52-62.

Hutchins, E. L., Hollan, J. D. and Norman, D. A. (1985) Direct Manipulation Interfaces. *Human-Computer Interaction*, **1**, 311-336.

Long, S., Aust, D., Abowd, G. D. and Atkeson, C. G. (1996) Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study, in *Proceedings of the 1996 conference on Human Factors in Computing Systems*, 293-294.

Long, S., Kooper, R., Abowd, G. D. and Atkeson, C. G. (1996) Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study, in *Proceedings of the 2nd Annual International Conference on Mobile Computing and Networking*.

Mankoff, J. and Abowd, G. D. (1997) Domisilica: Providing Ubiquitous Access to the Home, GVU Center, Georgia Institute of Technology.

Mankoff, J., Somers, J. and Abowd, G. D. (1998) Bringing People and Places Together, in *Proceedings of the 1998 Spring AAAI Symposium on Intelligent Environments*.

Minneman, S., Harrison, S., Janseen, B., Kurtenbach, G., Moran, T., Smith, I. and Melle, B. v. (1995) A Confederation of Tools for Capturing and Accessing Collaborative Activity, in *Proceedings of the ACM Conference on Multimedia 1995*.

Moran, T. P., Palen, L., Harrison, S., Chiu, P., Kimber, D., Minneman, S., Melle, W. v. and Zelweger, P. (1997) "I'll Get That Of the Audio'': A Case Study of Salvaging Multimedia Meeting Records, in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 202-209.

Mynatt, E. D., Back, M., Want, R. and Frederick, R. (1997) Audio Aura: Light-Weight Audio Augmented Reality, in *Proceedings of the ACM UIST'97 Symposium on User Interface Software and Technology*, 211-212.

Pandit, M. and Kalbag, S. (1997) The Selection Recognition Agent: Instant Access to Relevant Information and Operations, in *Proceedings of Intelligent User Interfaces '97*.

Picard, R. (1995) Affective Computing, MIT Media Lab, Perceptual Computing.

Pinkerton, M. (1997) Ubiquitous computing: Extending access to mobile data, GVU Center, Georgia Institute of Technology.

Schilit, W. N. (1995) System architecture for context-aware mobile computing, Columbia University.

Schneider-Hufschmidt, M., Kuehme, T. and Malinowski, U. (1993) Adaptive User Interfaces. North-Holland, Amsterdam.

Spreitzer, M. and Theimer, M. (1993) Scalable, Secure, Mobile Computing with Location Information. *Communications of the ACM,* **36(7)**.

Stifelman, L. J. (1996) Augmenting real-world objects: A paper-based audio notebook, in *Proceedings of the ACM 1996 Conference on Human Factors in Computing Systems*, 199-200.

Want, R., Hopper, A., Falcao, V. and Gibbons, J. (1992) The active badge location system. *ACM Transactions on Information Systems,* **10(1)**, 91-102.

Want, R., Schilit, B., Adams, N., Gold, R., Petersen, K., Ellis, J., Goldberg, D. and Weiser, M. (1995) The PARCTAB ubiquitous computing experiment, Xerox Palo Alto Research Center.

Weber, K. and Poon, A. (1994) Marquee: A tool for real-time video logging, in *Proceedings of the ACM 1994 Conference on Human Factors in Computing Systems*, 58-64.

Weiser, M. (1991) The Computer of the 21st Century. *Scientific American,* **265(3)**, 66-75.

Weiser, M. (1993) Some computer science issues in ubiquitous computing. *Communications of the ACM*, **7(36)**, 75-84.

Weiser, M. and Brown, J. S. (1997) Designing Calm Technology, in *Workshop on Ubiquitous Computing at CHI 1997*.

Whittaker, S., Hyland, P. and Wiley, M. (1994) Filochat: Handwritten notes provide access to recorded conversations, in *Proceedings of the ACM 1994 Conference on Human Factors in Computing Systems*, 271-277.

8 BIOGRAPHIES

Daniel Salber received his Ph.D. in Computer Science from the University of Grenoble in 1995. His research interests include ubiquitous computing and location-awareness, software engineering for interactive systems and mediaspace environments. He is currently a postdoc in the FCE group at Georgia Tech, funded by a fellowship from INRIA.

Anind K. Dey is a PhD student in the College of Computing and GVU Center at Georgia Tech. He received a BApSc in Computer Engineering from Simon Fraser University in 1993 and an MS in Aerospace Engineering from Georgia Tech in 1995. His research interests include the design of software infrastructures for supporting mobile and context-aware computing applications. He is currently being funded by Motorola on a University Partnership in Research award.

Gregory D. Abowd has been an Assistant Professor in the College of Computing and GVU Center at the Georgia Institute of Technology since 1994. His research interests include software engineering for interactive systems, with particular focus on future computing environments that emphasize mobile and ubiquitous computing applications. Dr. Abowd received a BS in Mathematics from the University of Notre Dame in 1986 and the degrees of M.Sc. (1987) and D.Phil. (1991) in Computation from the University of Oxford.