# Ubiquitous Computing: Past, Present and Future

Gregory D. Abowd and Elizabeth D. Mynatt

GVU Center and College of Computing
Georgia Tech
Atlanta, GA 30332-0280
*[abowd,mynatt]@cc.gatech.edu*

**Abstract**. The proliferation of computing into the physical world promises more than the ubiquitous availability of computing infrastructure; it suggests new paradigms of interaction inspired by constant access to information and computational capabilities. For the past decade, application-driven research in ubicomp has pushed three interaction themes: *natural interfaces*, *context-aware applications*, and *automated capture and access*. To chart a course for future research in ubiquitous computing, we review the accomplishments of these efforts and point to remaining research challenges.

Research in ubiquitous computing implicitly requires addressing some notion of scale; whether in the number and type of devices, the physical space of distributed computing or the number of people using a system. We posit a new area of application research, *everyday computing*, by focusing on scaling interaction with computing with respect to time. Just as pushing the availability of computing away from the traditional desktop fundamentally changes the relationship between humans and computers, providing *continuous* interaction moves computing from a localized tool to a constant companion. Designing for continuous interaction requires addressing interruption and resumption of interaction, representing passages of time and providing associative storage models.

Inherent in all of these interaction themes are difficult issues in the *social implications* of ubiquitous computing and the challenges of *evaluating* ubiquitous computing research. Although cumulative experience points to lessons in privacy, security, visibility and control, there are no simple guidelines for steering research efforts. Akin to any efforts involving new technologies, evaluation strategies form a spectrum from technology feasibility efforts to long-term use studies — but a user-centric perspective is always possible and necessary.

**Keywords:** human factors, design, user interfaces, ubiquitous computing, natural interfaces, context-aware applications, capture and access, augmented reality, evaluation, privacy, everyday computing, audio, vision, pen input

## 1 INTRODUCTION

Weiser introduced the area of ubiquitous computing (ubicomp) and put forth a vision of people and environments augmented with computational resources that provide information and services when and where desired (Weiser, 1991). For the past decade, ubicomp researchers have attempted this augmentation with the implicit goal of assisting everyday life and not overwhelming it. Weiser's vision described a proliferation of devices at varying scales, ranging in size from hand-held "inch-scale" personal devices to "yard-scale" shared devices. This proliferation of devices has indeed occurred, with commonly used devices such as the such as hand-held personal digital assistants (PDA), digitial tablets, laptops, and wall-sized electronic whiteboards. The development and deployment of necessary infrastructure to support continuous mobile computation is arriving.

Another aspect of Weiser's vision was that new applications would emerge that leverage off these devices and infrastructure. Indeed, ubicomp promises more than just infrastructure, suggesting new paradigms of interaction inspired by widespread access to information and computational capabilities. In this paper, we explore how this applications perspective has evolved in the decade since the start of the Ubiquitous Computing project at Xerox PARC. Specifically, we review the accomplishments and outline remaining challenges for three themes:

- We desire *natural interfaces* that facilitate a richer variety of communications capabilities between humans and computation. It is the goal of these natural interfaces to support common forms of human expression and leverage more of our implicit actions in the world. Previous efforts have focused on speech and pen input, but these interfaces still do not robustly handle the errors that naturally occur with these systems, and these interfaces are too difficult to build.
- Ubicomp applications need to be *context-aware*, adapting their behavior based on knowledge sensed from the physical and computational environment. Many applications have leveraged simple context, primarily location and identity, but numerous challenges remain in creating reusable representations of context, and in creating more complex context from sensor fusion and activity recognition.
- Finally, a large number of ubicomp applications strive to automate the capture of live experiences and provide flexible and universal access to those experiences later on. Several sophisticated systems for supporting meetings and classroom lectures have been created. Future efforts will undertake long-term personal capture systems as well as capture for distributed group activities.

Undertaking issues of scale is implicit in the definition of ubicomp research. Weiser defined the notion of scale as it incorporating a broad space of *computational devices* (Weiser, 1991). Likewise, scaling systems with respect to distribution of computation into *physical space* reinforces the desire to break the human away from desktop-bound interaction. Requirements for critical mass acceptance and collaboration imply scaling with respect to *people*. A final dimension, *time*, presents new challenges for scaling a system. Pushing the availability of interaction to a "24 by 7[1]" basis uncovers another class of largely unexplored interactions that will also push ubicomp research into the next century. To address scaling with respect to time, we introduce a new theme, called *everyday computing*, that promotes informal and unstructured activities typical of much of our everyday

---

[1] 24 hours a day, 7 days a week

Figure 1: Weiser (1991) described ubicomp devices of different size scales including the handheld PARCTab, the wearable active badge and the whiteboard sized Liveboard.

lives. These activities are continuous in time, a constant ebb and flow of action that has no clear starting or ending point. Familiar examples are orchestrating tasks, communicating with family and friends, and managing information.

As we review past work in ubicomp, an evolutionary path emerges. The first step, demonstrated by the PARCTab and Liveboad (see Figure 1), is computers encased in novel form factors. Often these compuational appliances push on traditional areas in computer science such as networking and operating systems. Since these new form factors often do not work well with traditional input devices such as the keyboard and mouse, developing new, and more natural, input capabilities is the next step. An example of this work is the pen-based shorthand language Unistroke for the PARCTab. After some initial demonstrations, infrastructure is needed to deploy these devices for general use. For example, numerous tour guide systems that mimic the first use of Active Badges (Want et al. 1992) have been built and deployed for real use.

It is at this point, not before, that application designers begin working with these new systems to develop novel uses, often focusing on implicit user input to minimize the intrusion of technology into everyday life. The objective of this application-centered research is to understand how everyday tasks can be better supported and how they are altered by the introduction of ubiquitous technologies. For example, ubicomp applications in support of common meeting tasks at PARC (through the Tivoli project) have resulted in new ways to scribe and organize materials during meetings. Capture environments in educational settings have provided more opportunities to understand the patterns of longer term reviewing tasks over large multimedia records. Applications of wearable computers initially emphasized constant access to traditional individual tasks, such as accessing e-mail. More recent applications have attempted to augment an individual’s memory and provide implicit information sharing between groups. The direction of applications research, what Weiser himself deemed the ultimate purpose for ubicomp research, is deeply influenced by authentic and extended use of ubicomp systems.

Today we are just starting to understand the implications of continuous immersion in computation. The future will hold much more than constant availability of tools to assist with whar are now traditional, computer-based tasks. Whether we wear computers on our body or have them embedded in our environment,[2] the ability of computers to alter our perception of the physical world, to support constant connectivity to distant people and places, to provide information at our fingertips, and to continuously partner with us in our thoughts and actions offers much more than a new "killer app" — it offers a killer existence. To examine a little more closely how applications research in ubicomp evolves, we will provide a brief history of uses for electronic whiteboards.

### 1.1 Case study: the electronic whiteboard

The history of the computer-augmented whiteboard illustrates the evolution of ubicomp in supporting everyday activities. As the first interactive, computer-augmented whiteboard, the LiveBoard (Elrod et al., 1992; Moran et al., 1996) was an inspiring demonstration of a computer in a novel form factor. It showcased great strides in building natural, pen-based interfaces. But the principal application domain of the LiveBoard, supporting meetings, was realized as a highly structured task requiring extensive pre- and post-processing of meeting information stored in a traditional hierarchical file system. Not all of this processing was automated, creating a burden on researchers that impacted how much authentic use experience could be obtained.

Following on from the seminal work on meeting capture, researchers at Georgia Tech developed a somewhat different application of the whiteboard for use in education. Work in the Classroom 2000 project (Abowd, 1999b) tailored the whiteboard for classroom lecturing and greatly automated the majority of pre- and post-processing work to provide capture of lectures for the duration of whole courses. Within the restricted application of pre-scheduled classroom lectures, a nearly walk-up and use system was created, and the captured material from the lectures was automatically made available to students via the Web. Serious engineering of the system resulted in the ability to capture nearly 100 different courses over the first four years of the project, leading to much deeper insight into the challenges of automated capture, which we will address later in this paper.

One of the positive results of the extended use of Classroom 2000 is that many users wanted to take advantage of the capture capabilities and natural whiteboard interaction in non-classroom situations. But the narrow domain of scheduled classroom lectures, the constraints of which made it possible to create a walk-up and use system, was too different from the serendipitous and informal types of meetings users wanted to also capture. This prompted the development of a whiteboard application that would support the capture of unplanned impromptu meetings centered around a shared writing surface (Brotherton et al., 1998). By simply sensing activity nearby the whiteboard, it became possible to capture discussions for unplanned groups of people. The ever-ready capture environment then created a problem for access. It was now necessary to use contextual information, such as who was present and the time of day, as cues to help access the informal discussions later on.

By focusing on whiteboard use in the individual office, Xerox PARC's Flatland (Mynatt et al, 1999) pushes ubicomp to support a range of everyday activities that are quite different from meeting or classroom whiteboard activities. First, the electronic whiteboard surface supports naive walk-up use, automatically clustering whiteboard content into segments and scaling segments to create more open space while maintaining their visibility. Second, instead of marshaling the whiteboard to support one task, such as running a meeting, Flatland users employ the whiteboard to support a

---

[2] Both is of course the most likely.

number of efforts in parallel, such as managing to-do items, drawing informal maps and performing simple calculations. The features of these applications, or behaviors, are tuned for quick illustrations, supporting office discussions and individual reflection, and are in sharp contrast to those typically available in production-oriented desktop software. Third, the whiteboard content is also automatically captured and tagged with relevant contextual information. By snapping to interesting points in a timeline for a whiteboard segment, users can easily browse, and then extend, past states of the whiteboard.

These latter two applications of electronic whiteboards, just scratched the surface in designing a continuously present whiteboard appliance to support common everyday tasks, impromptu discussions and the implicit capture and tagging of whiteboard material. Future efforts should focus on making the whiteboard an active partner instead of a well-designed tool as well as exploring constant connectivity to whiteboard material when physically distant. We'll return to these themes when we discuss *everyday computing.*

### 1.2 Overview

In this paper, we investigate the brief history of ubiquitous computing through exploration of the above-mentioned interaction themes—natural interfaces, context-aware computing, and automated capture and access for live experiences. In addition to reviewing the research accomplishments in these application research themes, we also outline some of of the remaining research challenges these application themes provide HCI researchers in the new millenium. We then explain the necessity for ubicomp research to explore continuous everyday activities. This motivates an area of research applications that builds off of the three earlier themes and moves ubicomp more into the realm of everyday computing characterized by continuously present, integrative, and unobtrusive interaction. Inherent in all of these interaction themes are difficult issues in the social implications of ubiquitous computing and the challenges of evaluating ubiquitous computing research. We conclude with our reflections on these issues and description, via case studies, of our current strategies for evaluation of ubicomp systems.

## 2 COMPUTING WITH NATURAL INTERFACES

Ubiquitous computing inspires application development that is "off the desktop." Implicit in this mantra is the assumption that physical interaction between humans and computation will be less like the current desktop keyboard/mouse/display paradigm and more like the way humans interact with the physical world. Humans speak, gesture and use writing utensils to communicate with other humans and alter physical artifacts. These natural actions can and should be used as explicit or implicit input to ubicomp systems.

Computer interfaces that support more natural human forms of communication (e.g. handwriting, speech, and gestures) are beginning to supplement or replace elements of the GUI interaction paradigm. These interfaces are lauded for their learnability and general ease of use, and their ability to support tasks such as authoring and drawing without drastically changing the structure of those tasks. Additionally, they can be used by people with disabilities for whom the traditional mouse and keyboard less accessible.

There has been work for many years in speech-related interfaces, and the emerging area of perceptual interfaces is being driven by a long-standing research community in computer vision and

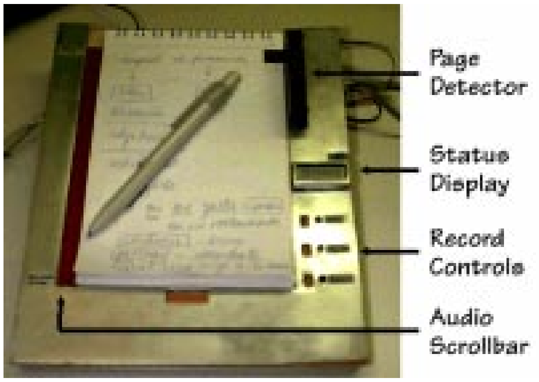

Figure 3: Stifelman's Audio Notebook (1996, 1997) is a good example of implicit natural input—the freehand notes act as indices into the captured audio.

computational perception (Perceptual UI, 1997,1998). Pen-based or free-form interaction is also realizing a resurgence after the failure of the first generation of pen computing. More recently, researchers have suggested techniques for using objects in the physical world to manipulate electronic artifacts, creating so-called graspable (Fitzmaurice et al, 1995) or tangible user interfaces (Ishii and Ullmer, 1997). Harrison et al. have attached sensors to computational devices in order to provide ways for physical manipulations of those devices to be interpreted appropriately by the applications running on those devices (Harrison et al, 1998). Applications that support natural interfaces will leverage off of all of these input and output modalities. Instead of attempting to review the impressive amount of work in natural interfaces, we focus on two issues that are important for enabling the rapid development of effective natural interfaces.

## 2.1 First-class natural data types

The landmark work of Douglas Engelbart and his team of researchers at SRI in the 60's demonstrated the power of building toolkits to bootstrap the development of increasingly sophisticated interactive systems. To ease the development of more applications with natural interfaces, we must be able to handle other forms of input as easily as keyboard and mouse input. The raw data or signals that underlie these natural interfaces, audio, video, ink, and sensor input, need to become first-class types in interactive system development. As programmers, we expect that any user interface toolkit for development provides a basic level of support for "fundamental" operations for textual manipulation, and primitives for keyboard and mouse interaction. Similarly, we need basic support for manipulating speech—such as providing speaker pause cues or selection of speech segments or speaker identification—as well as for video and ink and other signals, such as physical device manipulations detected by sensors.

Take, for example, freeform, pen-based interaction. Much of the interest in pen-based computing has focussed on recognition techniques to convert the "ink" from pen input to text. However, some applications, such as personal note-taking, do not require conversion from ink to text. In fact, it can be intrusive to the user to convert handwriting into some other form. Relatively little effort has been put into standardizing support for freeform, pen input. Some formats for exchanging pen input between platforms exist, but little effort has gone into defining effective mechanisms for manipulating the freeform ink data type within programs.

What kinds of operations should be supported for a natural data type such as ink? The Tivoli system provided basic support for creating ink data and distinguishing between uninterpreted, freeform ink data and special, implicitly structured gestures (Minneman et al, 1995; Moran et al, 1997, 1996). Another particularly useful feature of freeform ink is the ability to merge independent strokes together as they form letters, words and other segments of language. In producing Web-based notes in Classroom 2000 (discussed in more detail below), for example, we wanted annotations written with a pen by a lecturer to link to the audio or video of what was said or seen at that same time during a lecture (Abowd et al, 1998). The annotations are timestamped, but it is not all that useful to associate an individual penstroke to the exact time it was written in class. We used a temporal and spatial heuristic to statically merge penstrokes together and assign them a more meaningful, word-level time (Abowd, Brotherton and Bhalodia, 1998). Chiu and Wilcox (1998) have produced a more general and dynamic algorithm, based on hierarchical agglomeration, to selectively link audio and ink. These structuring techniques need to become standard and available to all applications developers who wish to create freeform, pen-based interfaces. And as the work of Chiu and Wilcox demonstrates, some of the structuring techniques can apply to more than one natural data type.

We must also think about primitive operations that combine different natural data types. Ever since the classic “Put-that-there” demonstration by Bolt (1980), we have had the promise of interaction techniques that smoothly mix different natural inputs. But this promise has not been realized in the form of improved paradigms or toolkits for building such multimodal dialogs.

### 2.2 Error-prone interaction for recognition-based interaction

When used for recognition-based tasks, natural interfaces come with a new set of problems—they permit new and more kinds of mistakes. When recognition errors occur, the initial reaction of system designers is to try to eliminate them, for example by improving recognition accuracy. However, Buskirk found that a reduction of 5-10% in the absolute error rate is necessary before the majority of people will even notice a difference in a speech recognition system (Buskirk, 1995).

Worse yet, eliminating errors may not be possible. Even humans make mistakes when dealing with these same forms of communication. As an example, consider handwriting recognition. Even the most expert handwriting recognizers (humans) can have a recognition accuracy as low as 54% when (Schomaker, 1994). Human accuracy increases to 88% for cursive handwriting (Schomaker, 1994), and 96.8% for printed handwriting (Frankish et al, 1995), but it is never perfect. This evidence all points to the conclusion that computer handwriting recognition will never be perfect.

Computer-based recognizers are even more error-prone than humans. The data they can use is often less fine-grained than what humans are able to sense. They have less processing power. And variables such as fatigue can cause usage data to differ significantly from training data, causing reduced recognition accuracy over time in recognition systems (Frankish et al, 1992).

On the other hand, recognition accuracy is not the only determinant of user satisfaction. Both the complexity of error recovery dialogues (Zajicek and Hewitt, 1990), and the value-added benefit for any given effort (Frankish et al, 1995), affect user satisfaction. For example, Frankish found that users were less frustrated by recognition errors when the task was to enter a command in a form than when they were writing journal entries. He suggests that the pay-back for entering a single word in the case of a command is much larger when compared with the effort of entering the word in a paragraph of a journal entry.

Error handling is not a new problem. In fact, it is endemic to the design of computer systems that attempt to mimic human abilities. Research in the area of error handling for recognition technologies must assume that errors will occur, and then answer questions about the best ways to deal with them. We see four key research areas for error handling of recognition-based interfaces (Mankoff and Abowd, 1999).

### 2.2.1 Error reduction

Error reduction involves research into improving recognition technology in order to eliminate or reduce errors. It has been the focus of extensive research, and could easily be the subject of a whole paper on its own. Evidence suggests that its holy grail, the elimination of errors, is probably not achievable. And large improvements (5-10%) are required before users even notice a difference (Buskirk, 1995).

### 2.2.2 Error discovery

Before either the system or the user can take any action related to a given error, one of them has to know that the error has occurred. The system may be told of an error through explicit user input, and can help the user to find errors through effective output of uncertain interpretations of recognized input. Three techniques are used to automate such error discovery—thresholding of confidence measures, historical statistics, and explicit rule specification.

Some recognition systems return a measure of the probability, or confidence measure, along with any interpretation of input. If this confidence measure is below some programmer-defined threshold, the system assumes the interpretation is incorrect and can display that assumption to the user.

When a recognition system does not directly return confidence measures, statistical analysis of historical data can be used to generate relevant probability measures. For example, Marx and Schmandt (Marx and Schmandt, 1994) compiled speech data about which letters were misrecognized as “e”, and with what frequencies. This statistical information was used to generate a list of potential alternatives whenever their speech recognizer returned “e”.

Baber and Hone (Baber and Hone, 1993) suggest using a rule base to determine when errors may have occurred. These rules can prove to be more sophisticated than either statistics or thresholding since they allow the use of context in determining whether an error has occurred. This analysis goes beyond simple statistics because it uses knowledge about the context in which a recognizer is being invoked. An example rule used in a handwriting recognition system for a programming environment might read:

```
“When the user has just written ‘for (‘, lower the probability of correctness
for any alternatives to the next word they write which are not members of the
set of variable names currently in scope.”
```

Some questions remain for this topic of error discovery. How can we improve the accuracy of error discovery? How should errors in error discovery be handled? What is the best technique for error discovery and how does this change depending on the situation?

### 2.2.3 Error correction techniques

Just as the user interface is the only way one party can inform the other that an error has occurred, it is also the only way that the user can correct an error. We found that current error handling techniques fall into three main categories—choosing a default, encouraging less ambiguous input, and mimicking natural human correction strategies.

The number of answers returned by an error-prone system is often larger than the number of answers expected by the user. This discrepancy leaves the interface designer with the choice of selecting none of the answers, or selecting one (or more) of the answers as “correct” by default. The designer can use information about the probability of correctness and the overhead for correcting a mistaken choice of default to decide when it is appropriate to choose a default.

A designer may choose to encourage a less ambiguous or error-prone input from the user. For example, the designers of the Palm Pilot™ force the user to write with a unistroke alphabet (Goldberg and Richardson, 1993). It is easier to recognize unistrokes than to recognize handwriting because there is no possibility of segmentation errors. In another example, Goldberg and Goodisman (1991) suggest using on-screen marks (boxes) to reduce segmentation errors and discourage cursive handwriting.

Humans have adopted strategies for correcting their own mistakes, and it is useful to mimic these strategies for correcting recognition errors. For example, when we mis-speak, we may pause, or repeat the correct word, with or without the addition of non-verbal audio cues to indicate the error. This kind of correction strategy is natural and promising for recognition systems. For example, Huerst et al. (1998) have experimented with a handwriting pre-processor which looks for handwriting correction marks, such as crossing out a letter, and applies these corrections before sending handwriting to a recognizer.

Some questions remain for error correction techniques. Does error correction require new types of interfaces or widgets different from other interfaces? When should error correction occur? How integrated should error correction interfaces be with the normal flow of an interface?

### *2.2.4 Toolkit level support*

Toolkits provide reusable components and are most useful when a class of common, similar problems exists. Interfaces for error handling would benefit tremendously from a toolkit that could be reused every time an error-prone situation arose. In addition to interface widgets, a toolkit would need to support complete reversibility, and keep track of multiple potential interpretations at once.

There is a serious question whether any of the discovery or correction techniques described above can be integrated into an event-based paradigm of current GUI toolkits. It is also not clear whether it is possible to separate out and encapsulate interface techniques for error handling. If such a reusable toolkit could be defined, how would we determine which discovery or correction techniques should be supported? Is complete reversibility possible, and if not what are the alternatives? Are there efficient ways of keeping track of increasing numbers of probabilities? We have begun to investigate the construction of such a reusable toolkit to support error recovery for recognition-based technologies (Mankof et al. 1999), but many problems remain to be tackled.

## 3 CONTEXT-AWARE COMPUTING

Two of the most compelling early demonstrations of ubicomp were the Olivetti Research Lab’s Active Badge (Want et al, 1992) and the Xerox PARCTab (Want et al, 1995), both location-aware appliances. These devices leverage a simple piece of context, user location, and provide valuable services (automatic call forwarding for a phone system, automatically updated maps of user

Figure 4: In Rekimoto and Nagao's NaviCam system (1995), a handheld device recognizes tagged objects and then overlays context-sensitive information.

locations in an office). Whereas the connection between computational devices and the physical world is not new—control systems and autonomously guided satellites and missiles are other examples—these simple location-aware appliances are perhaps the first demonstration of linking implicit human activity with computational services that serve to augment general human activity.

Numerous systems that leverage location as contextual information have been demonstrated. The most widespread applications have been GPS-based car navigation systems and handheld "tour guide" systems that vary the content displayed (video or audio) by a hand-held unit given the user's physical location in an exhibit area (Bederson, 1995; Abowd et al, 1997; Cheverst et al, 1998; Opperman and Specht, 1998).

Another important piece of context is recognizing individual objects. Earlier systems focused on recognizing some sort of barcode or identifying tag while recent work includes the use of vision-based recognition. Fitzmaurice (1993,1994) demonstrated using a hand-held device to "see inside" walls and pieces of machinery. Rekimoto and Nagao's NaviCam (1995, see Figure 4) recognized color barcodes overlaying additional information about objects on a hand-held video display. Recent efforts (Jebara, 1997) are attempting to substitute visual object recognition strategies so that objects do not have to be individually tagged.

Another twist on context-driven interaction is when the location is fixed but the user identity is variable. With passive or lightweight means of identifying users, interfaces can respond to the presence of people. Hudson and Smith (1996) created a "hot spot" along a corridor where users heard brief auditory summaries of their pending email, allowing them to be in contact with their desktop while physically away from their desk. As we discussed in the electronic whiteboard case study in Section 1.1, Brotherton et al (1999) demonstrated how a whiteboard in a common-area could automatically respond when multiple people gathered around it for an impromptu work session. The context of activity around the whiteboard automatic capture of the conversation and whiteboard interaction.

Although numerous systems that leverage a person's identity and/or location have been demonstrated, these systems are still difficult to implement. Salber et al. (1999) have created a "context toolkit" that simplifies designing, implementing and evolving context-aware applications.

This work emphasizes the strict separation of context sensing and storage from application-specific reaction to contextual information, and this separation facilitates the construction of context-aware applications. Mynatt (1998) points to the common design challenge of creating a believable experience with context-aware interfaces noting that the responsiveness of the interface is key to the person associating additional displays with their movements in the physical world.

In many ways, we have just scratched the surface of context-aware computing with many issues still to be addressed. Here we will discuss challenges in incorporating more context information, representing context, ubiquitous access to context sensing and context fusion, and the coupling of context and natural interaction to provide effective augmented reality.

### 3.1 What is context?

There is more to context than position and identity. Most context-aware systems still do not incorporate knowledge about time, history (recent or long past), other people than the user, as well as many other pieces of information often available in our environment. Although a complete definition of context is illusive, looking the “five W’s” of context we see:

**Who:** Current systems focus their interaction on the identity of one particular user rarely incorporating identity information about other people in the environment. As human beings, we tailor our activities and recall events from the past based on the presence of other people. Although systems have theorized on using the presence of others as an index into a captured history (Brotherton et al, 1999; Mynatt et al, 1999), not until these systems are actually used on a daily basis will the utility and privacy concerns of these ideas be understood.

**What:** The interaction in current systems either assumes what the user is doing or leaves the question open. Perceiving and interpreting human activity is a difficult problem. Nevertheless, interaction with continuously worn, context-driven devices will likely need to incorporate interpretations of human activity to be able to provide useful information. One strategy is to incorporate information about what a user is doing in the virtual realm. What application are they using? What information are they accessing? As a starting point, in the CyberDesk project, we used the context of what information the user was manipulating to provide automatic data integration between arbitrary tools (Dey et al, 1998). Another way of interpreting the “what” of context is to view it as the focus of attention of one or more people during a live event. Knowledge of the focus of attention at a live event can inform a better capture of that event, the topic of the next section of this paper.

**Where:** In many ways, the “where” question of context has been explored more than the others. Of particular interest, is coupling notions of “where” with other contextual information, such as “when.” Some tour guide systems have theorized about learning from a history of movements in the physical world, perhaps to tailor information display based on the perceived path of interest by the user. Again these ideas need fuller exploration.

**When:** With the exception of using time as an index into a captured record or summarizing how long a person has been at a particular location, most context-driven applications are unaware of the passage of time. Of particular interest is understanding relative changes in time as an aid for interpreting human activity. For example, brief visits at an exhibit could be indicative of a general lack of interest. Additionally, when a baseline of behavior can be established, action that violates a perceived pattern would be of particular interest. For example, a context-aware home might notice when an elderly parent deviated from a typically active morning routine, perhaps signaling an adult caregiver.

**Why:** Even more challenging than perceiving "what" a person is doing is understanding "why" they are doing it. Sensing other forms of contextual information that could give an indication of a person's affective state (Picard, 1997), such as body temperature, heartrate and galvanic skin response, may be a useful place to start.

### 3.2 Representations of context

Related to the definition of context is the question of how to represent context. This issue is important once we consider separating out the context-sensing portion of an application from its context-aware behavior, as suggested by Salber et al (1999). Without good representations for context, applications developers are left to develop ad-hoc and limited schemes for storing and manipulating this key information. The evolution of more sophisticated representations will enable a wider range of capabilities and a true separation of sensing context from the programmable reaction to that context.

### 3.3 The ubiquity of context sensing—context fusion

An obvious challenge of context-aware computing is making it truly ubiquitous. Having certain context, in particular positioning information, has been shown useful. However, there are few truly ubiquitous, single source context services. Positioning is a good example. GPS does not work indoors, and is even suspect in some urban regions as well. There are a variety of indoor positioning schemes as well, with differing characteristics in terms of cost, range, granularity, and requirements for tagging, and no single solution is likely to ever meet all requirements.

The solution for obtaining ubiquitous context is to assemble context information from a combination of related context services. Such *context fusion* is similar in intent to the related, and well-researched, area of sensor fusion. Context fusion must handle seamlessly handing off sensing responsibility between boundaries of different context services. Negotiation and resolution strategies need to integrate information from competing context services when the same piece of context is concurrently provided by more than one service. This fusion is also required because sensing technologies are not 100% reliable or deterministic. Combining measures from multiple sources could increase the confidence value for a particular interpretation. In short, context fusion assists in providing reliable ubiquitous context by combining services in parallel, to offset noise in the signal, and sequentially to provide greater coverage. For example, information from speaker identification, face recognition and foot-fall matching could be combined to identify an individual in a home setting. Information could be combined to increase reliability of identification. Additionally information from different sources may be available and more appropriate at different times.

The notion of ubiquitous context pushes harder on concerns for privacy and security of this kind of information. Our responsibilities as researchers include devising mechanisms for ensuring the security and veracity of the information obtained by context-aware applications, and designing appropriate applications that leverage this information for a greater perceived benefit in contrast to the costs incurred. As these issues cut across the whole of ubiquitous computing, we delve further into this discussion at the close of the paper.

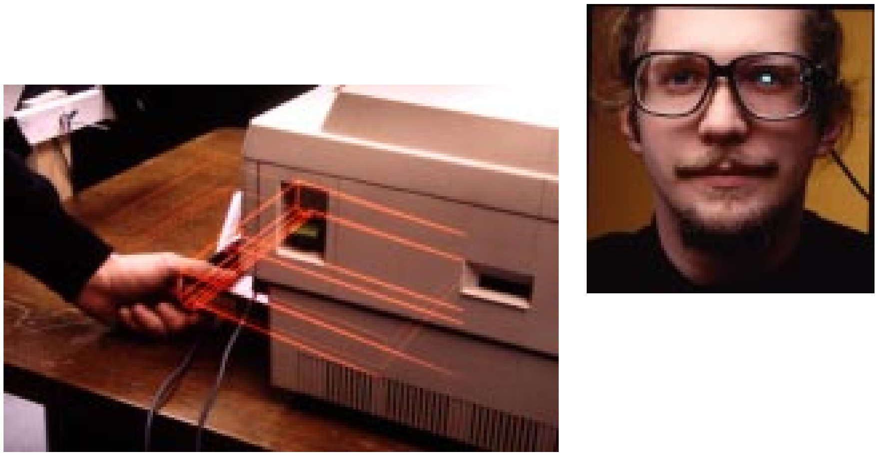

Figure 5: In the KARMA system (on the left) augmented views required heavy, clunky, head-mounted displays (Feiner, 1993). Now lightweight glasses, such as the ones shown on the right above from MicroOptical, provide similar display capabilities.

### 3.4 Coupling Context-Aware and Natural Interaction — Augmented Reality

The goal of many context-aware applications is to allow the user to receive, in "real-time," information based on actions in the physical world. The tour guide systems are a good example—the user's movements in an exhibit triggered the display of additional, context-sensitive information. These applications typically used separate, albeit portable, devices that required attention away from the rest of the physical world. The best metaphor to describe these interactions is that the user is "probing the world with a tool," similar to tools such as electronic stud-finders and geiger counters.

By incorporating augmented vision and augmented hearing displays, as well as natural input such as voice and gesture, we will be able to more closely integrate context-aware interaction with the physical world in which it resides (MacIntyre and Feiner, 1996, MacIntyre and Mynatt, 1998). In these interactions, the system is modifying how a user perceives the physical world. This tighter integration of information and perception should allow for more natural, seamless, hands-busy and serendipitous interaction. There are numerous technology and design issues to be solved to achieve this goal of context-sensitive perception of the physical world. Although visual displays are improving, especially with respect to the size and weight of the display device, they typically have a severely limited field of view. Our ability to layer sounds onto a dynamically changing background is limited and is much more art than science at this point.

## 4 AUTOMATED CAPTURE AND ACCESS TO LIVE EXPERIENCES

Much of our life in business and academia is spent listening to and recording, more or less accurately, the events that surround us, and then trying to remember the important pieces of information from those events. There is clear value, and potential danger, in using computational resources to augment the inefficiency of human record-taking, especially when there are multiple streams of related information that are virtually impossible to capture as a whole manually. Additionally, computational support can also automate explicit and implicit links between related, but separately generated, streams of information. A rich record of a group interaction can support later access to aid in recalling the meaning or significance of past events. It is even possible to use computational resources to assist in minimizing erroneous or embarrassing past events or, in contrast, to highlight particularly salient passages.

We need tools that seamlessly capture independent streams of information, automate the temporal and semantic relationships between those streams, and provide flexible, accessible and socially appropriate interfaces to the experience. Suites of these tools can remove the burden of doing something humans are not good at (i.e. recording) so that they can focus attention on activities they are good at (i.e. indicating relationships, summarizing, and interpreting).

There has been a good deal of research related to this general capture and access theme, particularly for meeting room/classroom environments and personal note-taking. We will focus our work on systems that handle capture for at least one natural data types as described in the previous section. This constraint rules out many of the generic capture/replay facilities, such as Lotus ScreenCam or macro recording capabilities available in desktop applications.

Early work by Schmandt & Arons (1985) and Hindus & Schmandt (1992) captured audio from phone conversations and provided ways to access the content of the recorded conversations. The two systems, PhoneSlave and Xcapture, treated audio as uninterpreted data and were successful using simple techniques to provide informative overviews of live conversations.

More recent research efforts have tried to capture other types of input, such as freeform ink written with a pen. Work at Xerox PARC resulted in the Tivoli system, a suite of tools to support a scribe at a meeting (Minneman et al, 1995, Moran et al, 1996), as well as some electronic whiteboard technology—the LiveBoard (Elrod et al, 1992)—to support group discussion. Artifacts produced on the electronic whiteboard during the meeting are timestamped. This temporal information is used after the meeting to index into recorded audio or video, thus providing the scribe a richer set of notes from the meeting. Similar integration between recorded ink annotations and audio/video is supported in Classroom 2000 for university lectures (Abowd et al, 1998a & 1998b, Abowd, 1999b), with a greater emphasis on automating the post-production of captured material into universally accessible interfaces for a large population of students. Figure 6 shows sample output from a Classroom 2000-supported lecture. Other capture systems, such as Authoring on the Fly (Bacher and Ottmann, 1996), and Cornell's Lecture Browser (Smith, 1999) also focus on capture of presentations with attention to capturing arbitrary program interactions and production-quality video capture from multiple sources.

The previous systems can be characterized as capturing the public portion of a live group experience. Prototypes such as Marquee (Weber, 1994), Filochat (Whittaker, 1994), We-Met (Rhyne and Wolf, 1992), the Audio Notebook (Stifelman, 1996, 1997}, Dynomite (Wilcox, 1997), NotePals (Davis et al, 1999), MRAS (White et al, 1998) and StuPad (Truong, 1999), all support individual annotation. Stifelman's Audio Notebook uses a particularly natural interface of pen and paper to produce a stenographer's notepad that automatically indexes each pen stroke to a digital audio record. StuPad demonstrates a capture system that works in concert with a group-oriented capture system, in this case the Classroom 2000 system. StuPad utilizes video tablet technology to provide a paper-like interface with electronically enhanced features, such as echoing of the lectures notes down to the student's seat. NotePals uses PDAs and paper-based electronic tablets (the

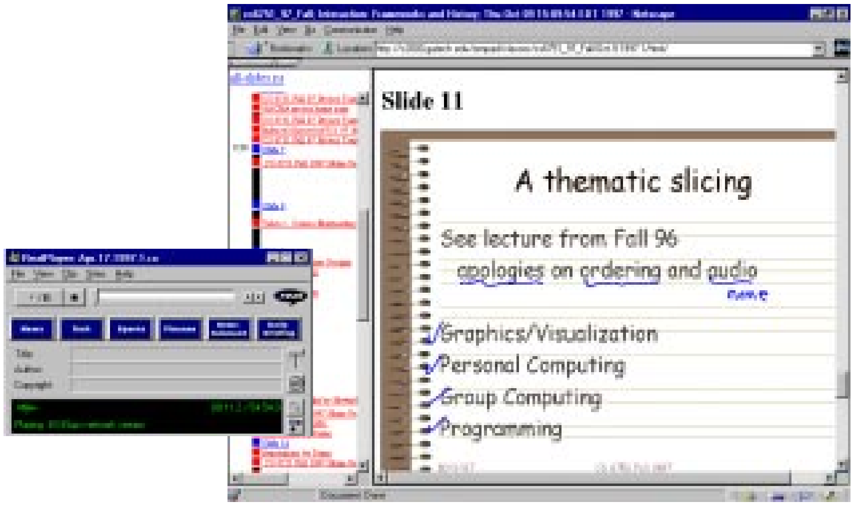

Figure 6: Classroom 2000 provides a web page for each captured lecture. Freehand pen strokes act as indices into the captured audio and video.

CrossPad™ from Cross Pen Computing) to provide group support for note-taking in educational or business settings.

Most of the above efforts produce some sort of multimedia interface to review the captured experience. By focusing on this post-production phase, some research endeavors provide automated support for multiple camera fusion, integration of various presentation media, and content-based retrieval mechanisms to help search through a large repository of captured information. The post-production results can then be accessed through a multimedia interface, typically distributed via the Web. Some research and commercial efforts include the Vicarious Learner from Edinburgh (Vicarious, 1999), Carnegie Mellon’s Just-In-Time Lectures (Just-In-Time, 1999), FlatLand from Microsoft Research (Bargeron et al, 1998), Bellcore’s AutoAuditorium (Bianchi, 1998), and CMU’s Informedia project (Informedia, WWW) as well as a number of commercial ventures, such as Eloquent, Inc. (Eloquent, WWW) and Islip (Islip, WWW).

In all of these cases, the emphasis on ubiquity is clearly seen in the separate capture and access phases. Electronic capture is moved away from traditional devices, like the keyboard, and brought closer to the user in the form of pen-based interfaces or actual pen and paper. Input in the form of voice and gesture is also accepted and is either treated as raw data or further interpreted to provide more understanding of the captured experience. Our own work in Classroom 2000 (Abowd, 1998) has focused on ubiquity of access, an issue of paramount importance in the educational domain. In our augmented classroom (see Figure 8), multiple streams of information (audio, video, annotated slides) are captured during a typical lecture. After a lecture, the system automatically weaves the captured events together into a set of standard HTML Web pages (see Figure 6).

## 4.1 Challenges in capture and access

Despite substantial research and advances in automated capture systems, there are a number of open research issues, that we summarize here. We separate out issues primarily associated with capture from those primarily associated with access.

### *4.1.1 Capture*

We have mentioned earlier the importance of having a good driving application for ubicomp research. In the capture domain, the main compelling applications have been for meeting support and education/training. These are indeed compelling application areas. In particular, our evidence in Classroom 2000 points to overwhelming acceptance of capture from the student perspective (Abowd, 1999b). There are many more possibilities, however, for exploring capture in equally compelling domains:

- Many of us record the special events in our lives—vacations, birthday parties, visits from relatives and friends—and we often spend time, years later, reflecting and remembering the events through the recordings on film and in diaries. How many times have we wished we had a camera at a particularly precious moment in our lives (a child’s first steps) only to fumble for the recording device and miss the moment? How difficult is it sometimes to find the picture or film of a significant event?
- In many collaborative design activities, the critical insights or decisions are often made in informal settings and are usually not documented properly. Technical exchanges often flow quite freely in opportunistic encounters. Even in more formal design meetings, the rich exchange of information and discussions around artifacts, such as storyboards or architectural recommendations, is often very poorly captured. Recently, we have begun experimenting with support to capture both informal brainstorming activities (Brotherton, 1999) and structured design meetings (Richter, 1999).
- Maintenance of a building might be better supported if we captured a record of the actual construction of the building—in contrast to the building plans. When repairs are needed, the appropriate technician could “replay” the construction and maintenance history of the relevant building artifact in order to determine the right course of repair.

To more fully understand how capture systems will change and augment daily practices, capture applications need to scale. For example, in Classroom 2000 we want to provide a student with a record of all lecture experiences during an academic career. To achieve this goal, the instrumented space of the individual classroom needs to be replicated in all locations on a campus that are likely to provide instruction. Classroom 2000 is the most extensively used capture system for education, but it is still a far way off from reaching the scale of an entire university campus. In order to reach this level of scale in space, a better understanding of the system and network-level requirements for a multimedia capture system is required (Padhye, 1997; Chervenak and Abowd, 1999). Such systems-level infrastructure needs to support synchronized capture of a wide variety of stream types (ink, audio, video, Web materials, presentations). There is evidence of this infrastructure emerging in some commercial products we can expect to see early in the next millennium.

With the exception of the Audio Notebook (Stifelman, 1997, thesis) and NotePals (Landay, 1998), there has been little work on capturing artifacts in the physical world and making them easily accessible in the access phase. For example, the Zen* system of Classroom 2000 supports capture of a presentation that has been prepared electronically, but what about capturing a presentation which exists only on paper? In order to make ubiquitous capture more transparent, the capture phase needs to more seamlessly handle physical as well as electronic artifacts.

Much of the capture currently being done is for what we would call *raw* streams of information that are captured mainly for the purpose of direct playback. No further analysis on those streams is done. However, it is often useful to *derive* additional information from a simple stream to provide a greater understanding of the live event. For example, Stifelman used results from discourse analysis to further segment the captured audio stream and make better prediction about when new topics commenced in a discussion (Stifelman, 1997). Similarly, Chiu and Wilcox proposed a hierarchical agglomeration technique for using pause detection to segment and associate both ink and audio (Chiu, 1998). Other computational perception techniques can be used to analyze the simple audio. ink or video signals.

Another application of signal analysis is to improve the recording of raw streams. How can we automate certain well-known production practices that merge multiple camera feeds into a single, coherent and high quality video that can be viewed later? Single, fixed camera angles are not sufficient to capture the salient parts of a live experience, but when we scale a system like Classroom 2000 to an entire campus, we cannot afford to pay technicians to sit in each of the classrooms. The single biggest challenge here is being able to determine the *focus of attention* for the group, and more difficult, for *each* individual at a live event.

### *4.1.2 Access*

In the access phase, we need to provide a number of playback capabilities. The simplest is to playback in realtime, but there are often situations in which this in inappropriate or overly inefficient. In reviewing a lecture for an exam, a student does not always want to sit through an entire lecture again, but he or she might want to pinpoint a particular topic of discussion and replay only that portion. Alternatively, a summarization of the experience which gleans salient points from across an entire captured session, might be more appropriate.

Synchronization of multiple captured streams during playback is vital. Commercial streaming products, such as RealNetworks G2/SMIL™ and Microsoft’s MediaPlayer/ASF™, are emerging standards to allow for powerful synchronization of programmer-defined media streams. However, it is not clear that any of these products will support the foreshadowing of streams so that a user can see what lies ahead in reviewing a stream. Such foreshadowing can help a user skim more quickly to a point of interest.

In most of the systems, the captured material is static upon reaching the access phase. Of course, there are often cases where annotating or revising captured material is appropriate, as well as then revising revised notes and so on. Although versioning is not a new problem to computer scientists, there are numerous challenges in providing an intuitive interface to multiple versions of captured material especially when some of the material is already time-based such as audio and video. A timeline is an effective interface for manipulating and browsing a captured session, but when the time associated with a captured artifact is split up into a number of non-contiguous time segments, the usefulness of the timeline is at least questionable. Newer time-based interaction techniques, such as Lifestreams (Fertig et al, 1996) or Timewarp (Edwards and Mynatt, 1997) are good starting points.

Finally, and perhaps most challenging, as these systems move from personalized systems to capturing events in more public settings, privacy concerns for the capture and later access of this material increases. Although these issues must be addressed in the specific design of each system, we still need general techniques for tagging material and authenticating access. We will discuss these issues later in this paper.

## 5 TOWARDS EVERYDAY COMPUTING

In this paper, we introduce a new area of interaction research, *everyday computing,* by focusing on scaling ubiquitous computing with respect to time. Just as pushing the availability of computing away from the traditional desktop fundamentally changes the relationship between humans and computers, providing *continuous* interaction moves computing from a localized tool to a constant presence. Our motivations for everyday computing stem from wanting to support the informal and unstructured activities typical of much of our everyday lives. These activities are continuous in time, a constant ebb and flow of action that has no clear starting or ending point. Familiar examples are orchestrating tasks, communicating with family and friends, and managing information.

Everyday computing leverages computational capabilities to enhance day-to-day *activities*. This focus on activities as opposed to tasks is crucial. The majority of computer applications support well-defined tasks that have a marked beginning and end with multiple subtasks in-between. Take word processing as an example. Word processing features are tuned for starting with a blank document (or a template), entering text, formatting, printing and saving. These applications are not well-suited to the more general activity of writing, encompassing multiple versions of documents where text is reused and content evolves over time.

Designing for everyday computing requires addressing these features of activities:

- They rarely have a clear beginning or end.

  Either as a fundamental activity, such as communication, or as a long-term endeavor, such as research in human-computer interaction, these activities have no point of closure. Information from the past is often recycled. Although new names may appear in an address book or new items on a to-do list, the basic activities of communication or information management do not cease.

  A basic tenet in HCI is designing for closure. Given a goal, such as spell-checking a document, the steps necessary to accomplish that goal should be intuitively ordered with the load on short-term memory held to a reasonable limit (Dix et al., 1998). The dialogue is constrained so that the goal is accomplished before the user begins the next endeavor. When designing for an activity, principles such as providing visibility of the current state, freedom in dialogue and overall simplicity in features play a more prominent role.

- Interruption is expected.

  Thinking of these activities as continuous, albeit possibly operating in the background, is a useful conceptualization. One side-effect is that resumption of an activity does not start at a consistent point, but is related to the state prior to interruption. Interaction must be modeled as a sequence of steps that will, at some point, be resumed and built upon. In addition to representing past interaction, the interface can remind the user of actions left uncompleted. As typically supported on many laptop computers, applications can make it easy to quickly *resume* the last state without lengthy time delays.

- Multiple activities operate concurrently.

Since these activities are continuous, the need for context-shifting amongst multiple activities is assumed. Application interfaces can allow the user to monitor a background activity, assisting the user in knowing when they should resume that activity. Resumption may be opportunistic, based on the availability of other people, or on the recent arrival of needed information. For example, users may want to resume an activity based on the number of related events that have transpired, such as reading messages in a newsgroup only after a reasonable number of messages have been previously posted. To design for background awareness, interfaces should support multiple levels of "intrusiveness" in conveying monitoring information that matches the relative urgency and importance of events.

Current desktop interfaces only provide a small beginnning in addressing these issues with multiple windows in a desktop interface. With minimal screen real estate, users must manage opening, closing and restacking the many windows associated with a variety of tasks. Simple awareness cues are included in some desktop icons, indicating that new email has been received for example, but there are few controls for creating levels of notification to meet different awareness needs.

The Rooms interface presented a compelling interface for spatially organizing documents and applications in multiple persistent working spaces (Henderson et al., 1986, Card et al. 1999). This standard has yet to be met by current commercial "task" bars for changing application focus. A useful extension to Rooms would be both to provide awareness of "background" rooms, and to assist the user in remembering their past activity when returning to a room.

- Time is an important discriminator.

Time is a fundamental human measuring stick although it is rarely represented in computer interfaces. Whether the last conversation with a family member was last week or five minutes ago is relevant when interpreting an incoming call from that person. When searching for a paper on a desk, whether it was last seen yesterday or last month informs the search.

There are numerous ways to incorporate time into human-computer interfaces. As we try to regain our working state, interfaces can represent past events contingent on the length of time (minutes, hours, days) since the last interaction. As applications interpret real-world events, such as deciding how to handle an incoming phone call or to react to the arrival at the local grocery store, they can utilize timing information to tailor their interaction.

- Associative models of information are needed.

Hierarchical models of information are a good match for well-defined tasks. In contrast, models of information for activities are principally associative since information is often reused on multiple occasions, from multiple perspectives. For example, assume you have been saving email from colleagues, friends and family for a long time. When dealing with current mail, you may attempt to organize it into a hierarchy of folders on various topics. Over time, this organization has likely changed, resulting in a morass of messages that can be searched with varying degrees of success. Likewise, interfaces for to-do lists are often failures given the difficulty in organizing items in well-defined lists.

Associative and context-rich models of organization support activities by allowing the user to re-acquire the information from numerous points of view. These views are inherent in the need to resume an activity in many ways, for many reasons. For example, users may want to retrieve information based on current context such as when someone enters their office or when they arrive at the grocery store. They may also remember information relative to other current information, for example, a document last edited some weeks ago or the document that a colleague circulated about some similar topic.

Of course, activities and tasks are not unrelated to each other. Often an activity will be comprised of a number of tasks, but the activity itself is more than these component parts for the reasons listed above. For example, communication activities contain well-defined tasks such as reading a message or composing a reply. The interaction falters when the task refers to the larger activity: How does this new message relate to previous messages from this person? What other issues should be included in the reply? The challenge in designing for activities is encompassing these tasks in an environment that supports continuous interaction.

### 5.1 Importance for Ubiquitous Computing

As computing becomes more ubiquitously available, it is imperative that the tools offered reflect their role in longer term activities. Although principles in everyday computing can be applied to desktop interfaces, these design challenges are most relevant given a continuously changing user context. In mobile scenarios, users shift between activities while the computing resources available to them also vary for different environments. Even in an office setting, various tools and objects play multiple roles for different activities. For example, use of a computer augmented whiteboard varies based on contextual information such as people present. Different physical objects such as a paper file or an ambient display can provide entry points and background information for activities. This distribution of interaction in the physical world is implicit in the notion of everyday computing, and thus clearly relevant to research in ubiquitous computing.

### 5.2 Synergy Amongst Themes

Research in everyday computing also continues to explore the three earlier interaction themes, but with the focus of designing a continuously available environment. Ishii's work in tangible media explores using natural interfaces to support communication and background awareness (Ishii et al, 1997). Current efforts in “Roomware” (Streitz, 1999) aim to create wall-sized and tablelike interaction areas that support a greater range of informal human activity.

With respect to context-aware interaction, Audio Aura (Mynatt, 1998) is clearly related to previous tour guide systems, as a change in location triggers information delivery on a portable device. The motivation for Audio Aura, however, is to continuously augment the background auditory periphery of the user. By adding dynamic information about the activity of colleagues and communication channels (e.g., email), Audio Aura enhances the perceptible sphere of information available while the user continues with daily activities.

Likewise, applications for automatic capture and indexing are moving into less structured environments. Starner and Rhodes (1997) describe a “Remembrance agent” that “remembers” information based on physical context information including visual recognition. As the user can instruct the system about what to remember, the agent becomes a storehouse of everyday information that is continuously available, but indexed based on physical location. A number of on-going research projects are exploring the design of “elephant boxes,” wearable devices that record continuously and later try to provide useful indices and summaries of the daily information they capture (Bush, 1945).

## 5.3 Research Directions in Everyday Computing

We posit that the field of everyday computing offers many challeneges to the HCI ressearch community. In our current and future work, we are focusing on the following questions:

- How to create the illusion of omnipresent computation?

  The current model for ubiquitous computing is one of constant availability. Since time is a fundamental human yardstick for understanding everyday events, we plan to shift this model to one of constant presence via time-sensitive interfaces. By sensing events in the user's physical world and correlating those events with actions on virtual devices such as today's desktop computers, we can create interfaces that help users manage current and future activity as well as recall and leverage past actions. Moreover, we can use these histories of interaction to help users regain their planned actions as they shift to a re-acquired activity. Exploring this time-centric interaction should help us understand how computers can be an effective partner in day-to-day activities.

- How to design visual and auditory cues for presenting information at different levels of the periphery of human attention?

  Despite increasing interest in tangible media and peripheral awareness, especially in computer-supported collaborate work (CSCW) and wearable computing, current interfaces typically present a generic peripheral backdrop with no mechanism for the user, or the background task, to move the peripheral information into the foreground of attention. Our current design experiments are aimed at creating peripheral interfaces that can operate at different levels of the user's periphery.

- How to connect events in the physical world to events in the virtual world and provide correlated information for interface displays and browsers?

  People operate in two disconnected spaces: the virtual space of email, documents and web pages and the physical space of face-to-face interactions, books and paper files. Yet human activity is coordinated across these two spaces. Despite efforts as early as the Digitial Desk (Wellner, 19xx) there is much work left to be done to understand how to combine information from these spaces to better match how people conceptualize their own endeavors.

- How to combine traditional HCI methods with in-the-field practices to support designing for informal, peripheral and opportunistic behavior?

  There is no one methodology for understanding the role of computers in our everyday lives. However combining information from methods as different as laboratory experiments and ethnographic observations is far from simple. In our research and classroom projects, our goal is to learn by doing, by interrogating the results we derive from different evaluation strategies. We have consciously chosen a spectrum of methods that we believe match the questions we are asking. Learning how these methods inform each other and how their results can be combined will be an on-going effort throughout our work. We continue this discussion in the next section on evaluating ubicomp systems.

# 6 ADDITIONAL CHALLENGES FOR UBICOMP

Two important topics for ubicomp research—evaluation and social implications—cut across all themes of research, and so we address them here.

## 6.1 Evaluating ubicomp systems

Evaluating ubicomp systems is difficult for several reasons, that we discuss below. After this brief discussion, we give specific advice for how to overcome these evaluation obstacles. We then demonstrate this advice through three separate case studies of ubicomp research, the AudioAura system, Xerox PARC's Flatland system, and Classroom 2000.

### *6.1.1 Evaluation challenges*

The first major difficulty in evaluating a ubicomp system is simply having a reliable system to evaluate. The technology used to create ubicomp systems is often on the cutting edge and not well understood by developers. Consequently, it is difficult to create reliable and robust systems that support some activity on a continuous basis. A good portion of reported ubicomp work remains at this level of unrobust demonstration prototyping. As we will show, it is still possible to do good user-centered feasibility research with these cutting-edge prototypes.

However, deeper evaluation requires real use of a system, and this, in turn, requires a deployment into an authentic setting. Furthermore, the scaling dimensions that characterize ubicomp systems—device, space, people or time—make it impossible to use traditional, contained usability laboratories. Effective evaluation, in which users are observed interacting with the system in routine ways, requires a realistic deployment into the environment of expected use.

Assuming an authentic deployment can occur, when users are comfortable with the service being provided and have developed habits for using the service, there is still the question of which qualitative or quantitative evaluation methods to apply. The majority of usability techniques are task-centric. If the user's tasks are known, then an evaluation is performed to determine the fitness of the system and its interface for completing that task. It is not at all clear how to apply task-centric evaluation techniques to activity-based everyday computing situations.

Another problem with task-centric evolution is the co-evolution of tasks and the computer artifacts built to support them (Carroll and Rosson, 1991). The mere act of introducing a new artifact to support some seemingly well-understood task changes the task itself. So even if we thought we knew how users would complete a simple task, such as keeping track of to-do items, providing a ubicomp system to continuously support the activity of to-do management would likely change the task. What this means is that a system must be observed in use over some period of time before an evaluator can understand the emergent tasks it supports and then evaluate those tasks.

### *6.1.2 How to overcome evaluation challenges*

It is not easy to overcome these evaluation challenges, and this is likely the reason we see relatively little published from an evaluation or end-user perspective in the ubicomp community. A notable exception is the work published by Xerox PARC researchers on the use of the Tivoli capture system in the context of technical meetings. Since research in ubiquitous computing will have limited impact in the HCI community until it respects the need for evaluation, we have some advice for those wishing to undertake the challenges.

Since authentic use of a system is a necessity, it is important in doing ubicomp research that a researcher build a compelling story, from the end-user's perspective, on how any system or infrastructure to be built will be used. The purpose of the compelling story is not simply to provide a demonstration vehicle for research results. It is to provide the basis for evaluating the impact of a system on the everyday life of its intended population. The best situation is to build the compelling story around activities that you are exposed to on a continuous basis. In this way, you can create a living laboratory for your work that continually motivates you to "support the story" and provides constant feedback that leads to better understanding of the use.

Designers of a system are not perfect, and mistakes will be made. Since it is already a difficult challenge to build robust ubicomp systems, you should not pay the price of building a sophisticated infrastructure only to find that it falls far short of addressing the goals set forth in the compelling story. You must do some sort of feasibility study of cutting-edge applications before sinking substantial effort into engineering a robust system that can be scrutinized with deeper evaluation. However, these feasibility evaluations must still be driven from an informed, user-centric perspective—the goal is to determine how a system is being used, what kinds of activities users are engaging in with the system, and whether the overall reactions are positive or negative. Answers to these questions will both inform future design as well as future evaluation plans. It is important to understand how a new system is used by its intended population before performing more quantitative studies on its impact.

*Case study: Audio Aura:* The affordances and usability issues of novel input and output technologies are not well understood when they are first introduced. Often these technologies are still unusable for any real, long-term use setting. Nevertheless user-centric evaluations are needed to influence subsequent designs. In the design of Audio Aura (Mynatt et al, 1998), we were interested in exploring how peripheral awareness of relevant office activities could be enhanced through use of ambient sound in a mobile setting, Our combination of active badges, wireless headphones and audio generation was too clunky for real adoption by long-term users. The headphones were socially prohibitive as they covered the ears with large, black shells. The capabilities for the development language, Java, to control sound presentation were too limited for creating rich auditory spaces. Nevertheless we wanted to understand the potential interaction knowing that these technological limitations would be removed in the future.

We employed *scenarios* of interaction, based on informal observations of the Xerox PARC work environment, to guide our design and evaluation. These scenarios incorporated information about how people at PARC work together, including practices such as gathering at the coffee bistro, often dropping by people's offices for impromptu conversations, and even the physical oddities of the building such as the long hallways that are the backbone of the layout. By grounding our scenarios in common practices, potential users could reflect on their daily activities when evaluating our designs. The scenarios also helped us understand a particular interaction issue: timing. In one of our scenarios, the communication path between the component technologies was not fast enough to meet the interaction demands. Although the speed could be increased, this modification required balancing a set of tradeoffs, namely speed vs. scalability, both important for our design goals. In short, the scenarios helped us understand the design space for further exploration.

Figure 7: In the design of Flatland, we used observations of whiteboard use to inform our design. Here whiteboard drawings on two different boards are used as the basis for discussing the concepts illustrated in the bottom slide.

*Case study: Xerox PARC's Flatland:* Designing ubiquitous computing applications require designers to project into the future how users will employ these new technologies. Although designing for a currently impossible interaction is not a new HCI problem, this issue is exacerbated by the implied paradigm shift in HCI resulting from the distribution of computing capabilities into the physical environment.

In our design work for Flatland (Mynatt et al, 1999), we employed ethnographic observations of whiteboard use in the office, coupled with questionnaires and interviews, to understand how people used their whiteboards on a daily basis. The richness of the data from the observations was both inspirational in our design work and a useful constraint. For example, the notion of "hot spots," portions of the board that users expect to change frequently, was the result of day-to-day observations of real whiteboard use. The data from the observations was key in grounding more in-depth user studies through questionnaires and interviews. Without this data, discussions would too easily slip into what users think they might do. By referring to two weeks of observational data, we were able to uncover and examine the details of daily practice.

Although the technology for our augmented whiteboard was not ready for deployment, or even user testing, we were able to gather a wealth of information from observations and interviews that critically informed our design.

*Case study: Classroom 2000:* In this last case study, centered at Georgia Tech, we demonstrate a much longer-term research project that evolved from early prototyping and feasibility studies into a more mature system that is currently used by a large population in a living classroom laboratory.

The Classroom 2000 project began in July 1995 with the intent of producing a system that would capture as much of the classroom experience as possible to facilitate later review by both students and teachers. In many lectures, students have their heads down, furiously writing down what they

Figure 8: In the Classroom 2000 project, we have had the ability to learn from long-term actual use in the Georgia Tech educational environment.

hear and see as a future reference. While some of this writing activity is useful as a processing cue for the student, we felt that it was desirable from the student and teacher perspective to afford the opportunity for students to lift their heads occasionally and engage in the lecture experience. The capture system was seen as a way to relieve some of the note-taking burden.

We needed to test the feasibility of this hypothesis quickly, so within six months of the project launch, we provided an environment to capture an entire course and observe whether our initial hypothesis was worth testing more vigorously. We learned some very valuable lessons during this first extended experience. The initial experiments included student note-taking devices that were clear distractions to the students (Abowd, 1999b), so we abandoned that part of the experiment, only to resume it in the past few months when the technology had caught up (Truong and Abowd, 1999).

We also learned from this initial experience that in order to understand the impact of this capture system on teaching and learning we would have to gather usage data from a larger set of classes. This required significant engineering effort to create a robust and reliable capture system that by the Spring Quarter of 1997 was able to support multiple classes simultaneously. Today, after capturing over 60 courses with 22 different instructors, we have gained significant insight into how the system is used and what future directions to take. Reports on numerous qualitative and quantitative studies have been published elsewhere (Abowd et al, 1998, 1999b; Chervenak et al, 1999; Brotherton and Abowd, 1999). As a direct result of these deeper evaluations, we know that the system encourages 60% of its users to modify their in-class note-taking behavior. We also know that not all of this modified behavior is for the better. Taking no notes, for example, is not a good learning practice to reinforce. We know that it is time to reintroduce student note-taking units that can personalize the capture experience and also encourage better note-taking practices. We also know to facilitate more content-based retrieval and synchronized playback of the lecture experience. We are exploring these paths now.

### 6.2 Social Issues for Ubiquitous Computing

We are pushing toward making it easier for computation to sense, understand and react to phenomenon in the physical world and to record those phenomena. These enabling technologies carry with them numerous dangers, for example making it too easy for people to build systems that effectively spy on others without any controlling authority. Ubicomp researchers would be remiss if they undertook their work without understanding these issues. However, the fear of wrong-doing is not a call to cease all work in this area, but to work toward technological, design and social solutions to address these concerns.

A basic concern about any information stored in a computer is knowing who can access and modify the contents. Where are the bits? Are they secure? *Security* and encryption schemes are part of the technological solutions available especially as information is gleaned from the environment and transported over networks. Alternatively, work in wearable computing emphasizes a design approach—providing security by keeping the bits local (on the body) and removing the risks of transporting them over a public network.

One fear of users is the lack of knowledge of what some computing system is doing, or that something is being done “behind their backs.” Although the original vision of ubiquitous computing described computing as disappearing into the physical environment, this “invisibility” is counter to informing users about how they are being sensed. To assuage that fear, design solutions can be employed to make this information *visible*. For example, systems that sense physical phenomena and capture live situations should provide clear indicators that this sensing or recording is occurring. As these sensing and recording capabilities are more commonly found, one challenge for everyday computing is to enable people to be aware of how they are being sensed. Just as people can ascertain their visibility in physical space (e.g How public is this space? Are there windows?) we need cues to convey our visibility in virtual space.

The next step is to allow those being sensed or recorded to have *control* to either stop this activity or to at least control the distribution and use of the information. This challenge is related to the design of collaborative environments where the actions and roles of collaborators are fluid and difficult to articulate in a static snapshot. The capture, distribution and use of information will be determined over time by the specific practices of people is workplace and home settings.

There are a number of reactions that system builders can have for handling the sensitive topic of when and what to capture. At Xerox PARC, one solution for capture was to agree to only capture the summary portions of technical meetings. In Classroom 2000, we defaulted to recording all of a lecture, but did not attempt to obtain good quality audio or video of anyone except the lecturer in the front of the room. In the Dynomite system from FX-PAL, the note-taker exercised control over which portions of the audio recording could be kept for future reference (Wilcox et al, 1997). Though this last solution was first presented as a means of reducing the amount of storage requirements for high-fidelity audio, we see merit in this approach from an individual’s perspective to enable an otherwise perfect capture system to “forget” some part of the past. An interesting challenge for collaborative situations is to figure out acceptable policies for “erasing” or “forgetting” some shared memory. A more positive slant on this issue would focus on ways to accommodate heightening awareness of particularly valuable segments of a captured experience instead of eliminating or forgetting parts of a captured history.

Although issues surrounding the appropriate use and dissemination of information are as old as the dawn of human communication, specific concerns stem from ubicomp making a new kind of information more generally available. The fact that computers can easily track our daily activities—a feat that previously required a large amount of human effort—is disconcerting at the least. In addition to addressing the above mentioned concerns of security, visibility and control, our approach is to create designed examples of appropriate and beneficial uses of this information. For example, one affordance of the low quality video in early media spaces was that the amount of information conveyed was more socially appropriate. The not-real-time, grainy images met important needs for awareness and feelings of connectivity without violating privacy concerns. In the design of Audio Aura (Mynatt et al, 1998), we took great care in conveying *qualitative* information about the activities of colleagues. When stopping by someone's office, information that could be obtained by the system (e.g. this person has not been in there office for a few hours) was akin to the information that someone in a neighboring office could provide.

There are other social issues as well that are not as directly linked to privacy. For example, recording a meeting or a lecture can have both positive and negative impact on those in attendance. On the positive side, knowledge of recording encourages people to be less reckless in their commentary. On the negative side, this same knowledge can cause people to refuse to contribute to a discussion for fear of saying something that would be regretted in the future. A more subtle problem was noticed in our extensive experience in Classroom 2000. Some students indicated that they chose not to ask questions in class because the answer was likely already discussed and it was up to the student to go back and listen to the lecture.

In general, social and legal practices continue to evolve in concert with technological and design innovations. In each situation people will compare the perceived benefits and costs of the uses of ubicomp technologies. For example, skiers and hikers choose to wear radio transponders so they can be located by rescue personnel. Firefighters benefit from understanding what each of them is doing and where they are located. Recent research details the calendaring practices at Sun Microsystem (Grudin et al, 1998) where colleagues share extensive information about their daily collaborative activities. As discussed in the previous section on evaluation, our understanding of the social implications of these technologies will often come after people invent new, unforeseen, uses of these technologies. Although the sand is always shifting beneath us, attention to issues of security, visibility, control and privacy should help create a brighter future.

## 7 CONCLUDING THOUGHTS

In this paper, we have attempted to outline the trajectory of ubicomp research in the decade since the inspiring work of Weiser and colleagues at Xerox PARC. We have identified three common themes for applications research in ubicomp, provided some background on significant achievements in those areas as well as highlighted some of the remaining challenges. We have done this with the desire to motivate budding ubicomp researchers to attack some important and well-defined problems. We no doubt have left out some other important challenges for ubicomp research, and we look forward to seeing those problems articulated and solved by others.

Weiser (1993) claimed that the whole point of ubiquitous computing was to create compelling applications that would drive the development of devices and infrastructure. We agree in spirit with this claim, but want to promote a broader view that promotes the general purpose utility (and challenge) of ubiquitous interaction with computational resources. The application or task-centric focus has been a fruitful one for HCI research. If we look at successful computing technology, however, it is not the case that a single application has driven critical mass acceptance and deployment. What is the motivating application for the personal computer in our office or home, or for a Palm Pilot? There are many applications, different for each person. The real goal for ubicomp is to provide many single-activity interactions that together promote a unified and continuous interaction between humans and computational services. The focus for the human at any one time is not a single interface to accomplish some task. Rather, the interaction is more free-flowing and integrative, akin to our interaction with the rich physical world of people, places and objects in our everyday lives.

## 8 ACKNOWLEDGMENTS

The authors would like to acknowledge the significant inspiration from the late Mark Weiser, who died somewhat suddenly during the writing of this paper. Mark's vision and enthusiasm has influenced a countless number of researchers, and we are proud to be associated to his vision for ubiquitous computing. We would also like to acknowledge the interactions and influence of a number of colleagues from Xerox PARC and Georgia Tech's Future Computing Environments Group within the College of Computing and GVU Center.f

## 9 REFERENCES

Abowd, G.D., C.G. Atkeson, J. Hong, S. Long, R. Kooper, M. Pinkerton (1997) ACM Wireless Networks, Vol. 3, pp. 421-433.

Abowd, G.D., Atkeson, C.G., Brotherton, J.A., Enqvist, T., Gully, P.A., Lemon, J. (1998a) Investigating the capture, integration and access problem of ubiquitous computing in an educational setting. Proceedings of CHI '98, PP. 440-447.

Abowd, G.D., J. Brotherton, and J. Bhalodia. (1998b) Automated capture, integration, and visualization of multiple media streams. In Proceedings of the 1998 IEEE conference on Multimedia and Computing Systems, 1998.

Abowd, G.D. (1999a) Software engineering issues for ubiquitous computing. In Proceedings of International Conference on Software Engineering (ICSE'99), Los Angeles, CA, pp. 75-84, May.

Abowd, G.D. (1999b) Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment. IBM Systems Journal, Special issue on Human-Computer Interaction: A focus on pervasive computing. Volume 38, Number 4. To appear.

Baber, C. and K.S. Hone, (1993) Modelling Error Recovery and Repair in Automatic Speech Recognition. International Journal of Man-Machine Studies, 39(3):495-515.

Bacher, C.H.R. and Ottmann, T.H. (1996) Tools and Services for Authoring on the Fly. In Proceedings of ED-MEDIA'96 Conference on Educational Multimedia and Hypermedia, Boston, MA, pp. 7-12, June.

Bargeron, D., A. Gupta, E. Sanocki and J. Grudin. (1998) Annotations for Streaming Video on the Web: System Design and Usage Studies. Microsoft Research Technical Report MSR-TR-98-60, Redmond, WA, November. Available at http://www.research.microsoft.com/scripts/pubDB/pubsasp.asp?RecordID=194.

Bederson, B. (1995) Audio Augmented Reality: A Prototype Automated Tour Guide In the Proceedings of ACM CHI'95, Short Paper, Vol.2, pp.210-211.

Bianchi, M.H. (1998) AutoAuditorium:A Fully Automatic, Multi-Camera System to Televise Auditorium Presentations. Presented at the 1998 Joint DARPA/NIST Smart Spaces Technology Workshop, 30-31 July 1998, National Institute of Standards and Technology, Gaithersburg, MD. Available at http://autoauditorium.bellcore.com/nist/autoaud.html.

Bolt, R.A. (1980) Put-that-there. Computer Graphics. Vol. 14, No. 3, pp. 262-270.

Brotherton, J.A., Abowd, G., and Truong, K.N. (1999) Supporting Capture and Access Interfaces for Informal and Opportunistic Meetings. GVU Technical Report, GIT-GVU-99-06, January 1999.

Bush, V. (1945) As We May Think. Atlantic Monthly, Volume 176, No. 1, July, pp. 101-108.

Buskirk, R.V. & LaLomia, M. (1995), The Just Noticeable Difference of Speech Recognition Accuracy, In Proceedings of ACM CHI'95 (Poster), p.95.

Card, S.K., Mackinlay, J., and Scheiderman, B. (1999) "Readings in Information Visualization: Using Vision to Think," Morgan Kaufmann, 1999.

Carroll, J.M. & Mary Beth Rosson (1991) Deliberated Evolution: Stalking the View Matcher in Design Space Articles, In *Human-Computer Interaction*, v.6 n.3-4 p.281-318.

Chervenak, A. L., V. Vellanki, I. Yanasak, N. Harel, R. Rodenstein, G.D. Abowd, J. Brotherton, and K. Ramachandran (1999) A Scalable Workload Model of Media-Enhanced Classrooms. Georgia Tech College of Computing Technical Report GIT-CC-99-06. February.

Cheverst K, K. Mitchell and N. Davies (1998) Design of an object model for a context sensitive tourist guide. In Proceedings of Interactive Applications of Mobile Computing - IMC'98, Rostock, Germany, November. Available at http://www.egd.igd.fhg.de/~imc98/proceedings.html.

Chiu, P. and L. Wilcox. (1998) A dynamic grouping technique for ink and audio. In Proceedings of UIST'98, pp. 195-202

Davis, R.C., J. A. Landay, V. Chen, J. Huang, R. B. Lee, F. C. Li, J. Lin, C. B. Morrey III, B. Schleimer, M. N. Price, and B. N. Schilit. (1999) NotePals: Lightweight Note Sharing by the Group, for the Group. In Proceedings of CHI'99, Pittsburgh, PA, May 1999. To appear.

Dey, A, Abowd, G.D and Wood, A. (1998) Cyberdesk: A Framework for Dynamic Integration of Desktop and Network-based applications. Knowledge-Based Systems. Vol. 11, pp. 3-13.

Dix, A., Finlay, J., Abowd, G., and Beale, R. (1998) *Human-Computer Interaction*, Second Edition, Prentice Hall.

Edwards, W.K. and E.D. Mynatt (1997) Timewarp: Techniques for Autonomous Collaboration. In Proceedings of ACM CHI'97, pp. 218-225.

Elrod, S., Bruce, S., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K, Tang, J., and Welch, B. (1992) Liveboard: A large interactive display supporting group meetings, presentations and remote collaboration. In Proceedings of CHI '92, pp. 599-607, May.

Eloquent, Inc. (1999) See www.eloquent.com.

Feiner, Steven, MacIntyre, Blair and Seligmann, Doree. ``Knowledge-Based Augmented Reality." In Communications of the ACM 36(7): 53-62, July, 1993.

Fertig, S., E. Freeman and D. Gelernter (1996) Lifestreams: An Alternative to the Desktop Metaphor. In Proceedings of ACM CHI'96, pp. 410-411.

Fitzmaurice, G.W., W. Buxton (1994) The Chameleon: Spatially Aware Palmtop Computers VIDEOS: Part II - Techniques for Improved Human-Computer Interaction Proceedings of ACM CHI'94 Vol. 2 p.451-452.

Fitzmaurice, G.W., H. Ishii, and W. Buxton (1995) Bricks: Laying the Foundations for Graspable User Interfaces. Proceedings of ACM CHI'95, May, p.442-449.

Fitzmaurice, G.W., S. Zhai and M.H. Chignell (1993) Virtual Reality for Palmtop Computers Special Issue on Virtual Worlds. ACM Transactions on Information Systems Vol.11 No. 3 p.197-218.

Frankish, C., Jones, D. & Hapeshi, K. (1992), Decline in Accuracy of Automatic Speech Recognition as Function of Time on Task: Fatigue or Voice Drift?, International Journal of Man-Machine Studies 36(6),797-816.

Frankish, C., Hull, R. & Morgan, P. (1995), Recognition Accuracy and User Acceptance of Pen Interfaces, In Proceedings of ACM CHI'95, pp.503-510.

Gelernter, D.H. (1999) *Machine Beauty: Elegance and the Heart of Technology*, Mastermind, 1999.

Goldberg. D. and A. Goodisman (1991), (STYLUS} User Interfaces for Manipulating Text. In Proceedings of ACM UIST'91, pp.127-135.

Goldberg, D. and Richardson, C. (1993), Touch-Typing with a Stylus, In Proceedings of ACM INTERCHI'93, pp. 80-87.

Harrison, B.J., K. P. Fishkin, A. Gujar, C. Mochon and R. Want. (1998) Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces. In Proceedings CHI'98, pages 17-24, May.

Henderson, J.D.A., and Card, S.K. (1986), "Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention in Window-based Graphical User Interfaces," *ACM Transactions on Graphics*, Vol. 5, No. 3, July 1986, pp. 211–241.

Hindus, D. and Schmandt, C. (1992) Ubiquitous audio: Capturing spontaneous collaboration. In Proceedings of CSCW'92, pp. 210-217, November.

Hudson, S. and I. Smith (1996) Electronic Mail Previews Using Non-Speech Audio. In Proceedings of ACM CHI'96, Short paper. v.2 p.237-238.

Huerst, W., J. Yang, and A. Waibel. (1998), Interactive Error Repair for an Online Handwriting Interface, In Proceedings of ACM CHI'98 (Poster), pp.353-354.

Informedia project (www.informedia.cs.cmu.edu}

Ishii, H. and B. Ullmer (1997) Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms. In Proceedings of CHI'97, pages 234-241, May.

Jebara, T. Cyrus Eyster, J. Weaver, T. Starner, and A. Pentland (1997), Stochasticks: Augmenting the billiards experience with probabilistic vision and wearable computers, In *IEEE International Symposium on Wearable Computing*, Oct. 1997.

Islip Media (1999) See http://www.islip.com.

Just-In-Time Lectures (1999) Carnegie Mellon University. See http://www.jitl.cs.cmu.edu.

MacIntyre, B., and Feiner, S. ``Future Multimedia User Interfaces." in Multimedia Systems, 4(5), 1996.

MacIntyre, B. and Mynatt, E.D. ``Augmenting Intelligent Environments: Augmented Reality as an Interface to Intelligent Environments." In AAAI 1998 Spring Symposium Series, Intelligent Environments Symposium, March 23-25, 1998, Stanford University, CA.

Mankoff, J. and G. Abowd. (1999) Error correction techniques for handwriting, speech and other ambiguous or error prone systems. Georgia Institute of Technology, GVU Center, Technical Report GIT-GVU-99-18.

Mankoff, J., G. Abowd and S. Hudson (1999) Error correction techniques for handwriting, speech, and other ambiguous or error prone systems. Georgia Institute of Technology, GVU Center, Technical Report GIT-GVU-99-26.

Marx, M. and C. Schmandt, (1994), Putting People First: Specifying proper names in speech interfaces. In Proceedings of ACM UIST'94, pp. 30-37.

Minneman, S., Harrison, S., Janseen, B., Kurtenbach, G., Moran, T., Smith, I. and van Melle, B. (1995) A confederation of tools for capturing and accessing collaborative activity. In Proceedings of Multimedia ‘95. November.

Moran, T. Chiu, P. and van Melle, B. (1996) Finding and using implicit structure in human-organized spatial layouts of information. In Proceedings of CHI’96, pages 346-353.

Moran, T., Chiu, P., Harrison, S., Kurtenbach, G., Minneman, S., and van Melle, W. (1996) Evolutionary engagement in an ongoing collaborative work process: A case study. In Proceedings of CSCW’96.

Moran, T.P., P. Chiu, and W. van Melle. (1997) Pen-based interaction techniques for organizing material on an electronic whiteboard. In Proceedings of the ACM Symposium on User Interface Software and Technology - UIST’97, pages 45-54. ACM Press, October.

Moran, T.P., L. Palen, S. Harrison, P. Chiu, D. Kimber, S. Minneman, W. van Melle and P. Zellweger (1997) “I’ll get that off the audio”: A case study in salvaging multimedia meeting records. In Proceedings of CHI’97.

Mynatt, E. D., Back, M., Want, R., Baer, M. and Ellis, J. (1998). “Designing Audio Aura,” in the Proceedings of the 1998 ACM Conference on Human Factors in Computing Systems (CHI’98)., Los Angeles, CA., 566- 573.

Mynatt, E.D., Igarashi, T., Edward, W.K., and LaMarca, A. (1999) “Flatland: New Dimensions in Office Whiteboards,” to appear in the 1999 ACM Conference on Human Factors in Computing Systems.

Mynatt, E.D. (1999) The Writing on the Wall. In Human-Computer Interaction (INTERACT ‘99). Angela Sasse and Chris Johnson (Editors). Published by IOS Press, IFIP TC.13, 1999.

Oppermann, R. and M. Specht (1998) Adaptive support for a mobile museum guide. In Proceedings of Interactive Applications of Mobile Computing - IMC’98, Rostock, Germany, November. Available at http://www.egd.igd.fhg.de/~imc98/proceedings.html.

Padhye, J. and J. Kurose. (1997) An Empirical Study of Client Interactions with a Continuous-Media Courseware Server. University of Massachusetts Technical Report UM-CS-1997-056.

Perceptual User Interfaces Workshop - PUI’97. (1997) Banff, Alberta, Canada, October 20-21. http://research.microsoft.com/PUIWorkshop97/.

Perceptual User Interfaces Workshop - PUI’98. (1998) San Francisco, Nov. 5-6. http://research.microsoft.com/PUIWorkshop/PUI98.htm.

Picard, R. (1997) Affective Computing. MIT Press.

Rekimoto, J. and K. Nagao (1995) The World through the Computer: Computer Augmented Interaction with Real World Environments Virtual and Augmented Realities. In the Proceedings of the ACM Symposium on User Interface Software and Technology - UIST’95, p.29-36.

Rhodes, B.J. (1997) “The Wearable Remembrance Agent: A system for augmented memory,” in Personal Technologies Journal Special Issue on Wearable Computing, Personal Technologies 1(4), pp. 218-224.

Richter, H., P. Schuchhard and G.D. Abowd. (1999) Automated capture and retrieval of architectural rationale. Technical note to appear in on-line proceedings of the First Working IFIP Conference on Software Architecture, San Antonio, February.

Salber, D., A.K. Dey and G.D. Abowd (1999) The Context Toolkit: Aiding the Development of Context-Enabled Applications. In Proceedings of ACM CHI’99, Pittsburgh, PA, May 15-20, To appear.

Schmandt, C. and B. Arons. PhoneSlave: A graphical telecommunications interface. In Proceedings of the Society for Information Display, 26(1):79-82,

Schomaker, L. R.B. (1994), User-interface Aspects in Recognizing Connected-Cursive Handwriting, In Proceedings of the IEE Colloquium on handwriting and Pen-based input, Number 1994/065, The Institution of Electrical Engineers, London.

Smith, B. (1999) Cornell's Lecture Browser.

Starner, T. S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R. Picard, and A. Pentland (1997), Augmented Reality Through Wearable Computing, In *Presence* 6(4).

Stifelman, L.J. (1996) Augmenting real-world objects: A paper-based audio notebook. In Proceedings of CHI'96, pp. 199-200. April.

Stifelman, L.J. (1997) The Audio Notebook: Paper and Pen Interaction with Structured Speech. Ph.D. thesis, MIT Media Lab, August.

Streitz, N.A., Jorg Gei ler, Torsten Holmer, Shin'ichi Konomi, Christian Muller-Tomfelde, Petra Rexroth, Peter Seitz, Wolfgang Reischl, Ralf Steinmetz (1999) i-Land: An Interactive Landscape for Creativity and Innovation, to appear in the 1999 ACM Conference on Human Factors in Computing Systems.

Truong, K.N. and G.D. Abowd. (1999) StuPad: Integrating student notes with class lectures. Submitted as late-breaking result to CHI'99. January.

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

Want, R., B. Schilit, N. Adams, R. Gold, K. Petersen, J. Ellis, D. Goldberg and M. Weiser. (1995) The PARCTab ubiquitous computing experiment. Xerox Palo Alto Research Center, Technical Report CSL-95-1, March.

Weber, K and Poon, A. (1994) A tool for real-time video logging. In Proceedings of CHI'94, pp. 58-64, April.

Weiser, M. (1991) The computer of the 21st century. Scientific American, 265(3):66-75, September.

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

White, S.A., A. Gupta, J. Grudin, H. Chesley, G. Kimberly, and E. Sanocki (1998) A Software System for Education at a Distance: Case Study Results. Microsoft Research Technical Report MSR-TR-98-61, Redmond, WA, November. Available at http://www.research.microsoft.com/scripts/pubDB/pubsasp.asp?RecordID=198

Whittaker, W., Hyland, P. and Wiley, M. (1994) Filochat: Handwritten notes provide access to recorded conversations. In Proceedings of CHI'94, pp. 271-277, April.

Wilcox, L., B. Schilit and N. Sawhney. (1997) Dynomite: A dynamically organized ink and audio notebook. In Proceedings of CHI'97, pages 186-193.

Zajicek, M. & Hewitt, J. (1990), An Investigation into the Use of Error Recovery Dialogues in a User Interface Management System for Speech Recognition, In Proceedings of IFIP INTERACT'90, pp.755-760.

The Vicarious Learner Project. (1999) University of Edinburgh. See www.hcrc.ed.ac.uk/gal/vicar.

Wolf, C. and Rhyne, J. (1992) Communication and information retrieval with a pen-based meeting support tool. In Proceedings of CSCW'92, pp. 322-329.