UBIQUITOUS COMPUTING: RESEARCH THEMES AND OPEN ISSUES FROM AN APPLICATIONS PERSPECTIVE

Gregory D. Abowd

GVU Center & College of Computing
Georgia Institute of Technology, Atlanta, GA, USA
+1 404 894 7512
abowd@cc.gatech.edu or http://www.cc.gatech.edu/fce

ABSTRACT

The defining characteristic of ubiquitous computing is the attempt to break away from the traditional desktop computing paradigm and move computational power into the environment that surrounds the user. Applications of ubiquitous computing technology are the main drivers for research in this area. We describe the work over the past 18 months done by the Future Computing Environments Group at Georgia Tech. We have an applications perspective on ubiquitous computing and have prototyped many systems using a wide range of technology in several separate domains. This paper summarizes three emergent research themes that are the result of generalizing our prototyping and evaluation experience: automated capture, integration and access; context-awareness; and ubiquitous software services. We define each of these themes, demonstrate systems we have built which emphasize the theme, and present a number of open issues that will guide our future work and hopefully that of others. We conclude by sharing a number of general insights on our research method in ubiquitous computing.

KEYWORDS: ubiquitous computing, context-aware computing, mobile computing, automated information capture and access, scalable interfaces, software service integration

INTRODUCTION

The interest in ubiquitous computing has surged over the past few years, thanks to some influential writings and plenty of experimental work. The defining characteristic of ubiquitous computing is the attempt to break away from the traditional desktop interaction paradigm and move computational power into the environment that surrounds the user. Rather than force the user to search out and find computer's interface, ubiquitous computing suggests that the interface itself can take on the responsibility of locating and serving the user.

Full paper submitted to CHI'97

The history of computing is filled with examples of radical paradigm shifts in the way humans interact with and perceive technology (see [17] or [6, Ch. 4] for more complete historical accounts). From the introduction of video display units to the separate inspirations of networked time-sharing, computer graphics and personal computing, the ideas of a few creative people have drastically modified, and for the most part augmented (the arguments presented by Landauer [10] notwithstanding), our own capabilities. The vision of ubiquitous computing —first expressed by Weiser [22] and grounded in experimental work done at Xerox PARC— holds the promise of yet another interaction paradigm shift.

What is ubiquitous computing technology? Our general working definition is any computing technology that permits human interaction away from a single workstation. This includes pen-based technology, hand-held or portable devices, large-scale interactive screens, wireless networking infrastructure, and voice or vision technology.

As Weiser points out, "Applications are of course the whole point of ubiquitous computing." [23] Fueled by this statement, and somewhat perplexed by research in mobile computing that is practically void of any applications focus, [1] we initiated a group at Georgia Tech, the Future Computing Environments (FCE) Group, to investigate and invent applications of ubiquitous and mobile computing technology. In this paper, we will describe some of the projects undertaken by the FCE group since April of 1995. Our research method is to identify application domains that can benefit from ubiquitous technology, to produce prototype systems rapidly (within 3-6 months), and to exercise and evaluate those systems in real-life situations. This method has resulted in a number of experimental systems, a lot of war stories of late-night panic attacks, and some insights into general research themes that have emerged as challenges

[1] peruse recent proceedings and edited texts on mobile computing that are mostly dominated by network and operating systems researchers [4, 9]

for ubiquitous computing applications. We will spare the reader from the war stories and share the insights within the context of various projects.

We have identified three principle and emergent research themes from our experience:

- automating the capture of individual and group experiences in order to facilitate access to a richly integrated record of events;
- customizing computational services based on knowledge of the user's physical context; and
- providing ubiquitous software services that are available on many different physical devices and integrated with a changing set of other services.

In the remainder of this paper, we address each of these themes in turn. We first define what the problem is and provide some background information. We demonstrate our applications perspective on the theme by describing one or more projects of working systems that highlight the importance of that particular theme. Finally, we identify some open issues relating to the theme, some that are the focus of our near-term research and some that we hope will motivate other colleagues.

AUTOMATED CAPTURE, INTEGRATION, AND ACCESS

Defining the problem

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

There has been a good deal of research related to this general CIA theme, particularly for meeting room environments and personal note-taking. Work at Xerox PARC has resulted in a suite tools to support a sribe at a meeting [14, 15], as well as some electronic whiteboard technology —the LiveBoard [7]— to support group discussion. The Marquee note-taking system from PARC [21] and the Filochat prototype at Hewlett-Packard Labs [24] both supported individual annotation. A simple pen-based interface to produced automatic indexes into either a video (for Marquee) or an audio (for Filochat) stream that could be be traversed later on during access and review. Stifelman used an even more natural interface of pen and paper to produce a stenographer's notepad that automatically indexed each penstroke to a digital audio record [19]. The implicit connection between the note-taking device and alternate information streams (audio and/or video) is a common theme that has also been explored at MIT's Media Lab [8] and at Apple [5].

In all of these cases, the emphasis on ubiquity is clearly seen in the capture and integration phases. Electronic capture is moved away from traditional devices like the keyboard and brouht closer to the user in the form of pen-based interfaces or actual pen and paper. There is not so much emphasis on ubiquity of access, however, and our own work was performed in a domain — education— in which ubiquitous access was of paramount importance.

Our experience: Classroom 2000

As we are situated in an academic environment, it was natural for us to ask what effect ubiquitous computing could have in the classroom. Were pen-based interfaces for a student and an entire classroom something that would radically change the way in which we teach and learn? We initiated the Classroom 2000 project to investigate this issue.

One way to view classroom teaching and learning is as a group multimedia authoring activity. Before class, teachers prepare outlines, slides, or notes and students read textbooks or other assigned readings. During the lecture, the words and actions of the teacher and students expound and clarify the lessons underlying the prepared materials. It is common practice to annotate the prepared material during the lecture and to create new material as notes on a whiteboard or in a student notebook. These different forms of material —printed, written and spoken— are all related to the learning experience that defines a particular course, and yet there are virtually no facilities provided to automatically record and preserve the relationships between them. Applying a variety of technology —electronic whiteboards, personal pen-based interfaces, digital audio and video recording, and the World-Wide Web— would allow us to test whether ubiquitous computing positively affects the teaching and learning experience.

Classroom 2000 is fundamentally a CIA problem with the principle distinction from previous work being the application in an educational setting with many simultaneous scribes (teacher and all students). Though our long-term goal is to be able to provide CIA support for all varieties of teaching and learning styles, it was clear early on that we would not be able to develop initial prototypes of much generality. We have developed a number of prototype systems to suit varying teaching and learning styles [2]. The system we describe here suited a teaching style that is similar to a conference presentation. The teacher prepares slides and gives copies to the students before class. Students can take notes on top of the slides and the teacher displays the slides and annotates them during the lecture.

On the left of Figure 1 is a screenshot of the note-taking application, called ClassPad, a Visual Basic application

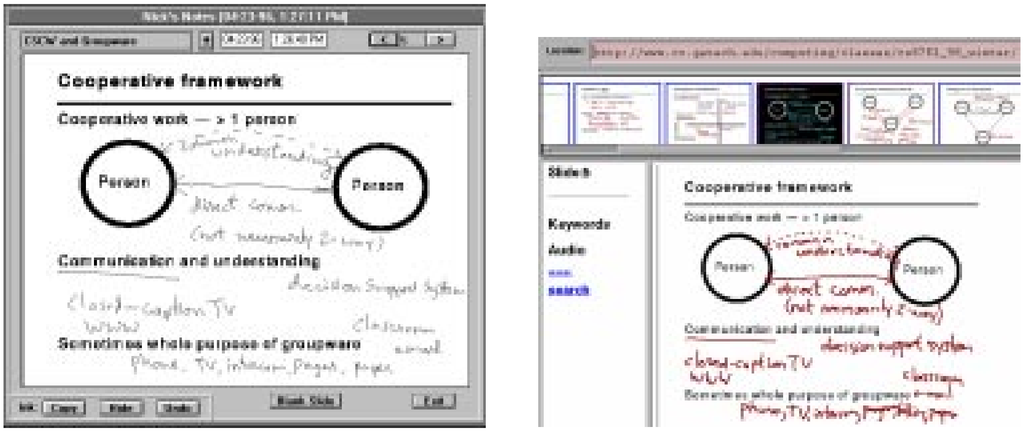

Figure 1: Classroom 2000 screenshots. The left shows the interface for ClassPad, a Visual Basic note-taking prototype. This same interface served as the interface for the teacher's electronic whiteboard. On the right is a sample of the automatically-generated audio-enhanced and searchable Web notes.

that runs on any Windows 3.x pen-based platform supporting Pen for Windows 1.0. In our experiments, we used a LiveBoard for the teacher and a variety of pen-based PCs for the student notebooks (Dauphin DTR-1, Toshiba DynaPad T100, and IBM ThinkPad 730T machines). ClassPad allows students to annotate on top of the teacher's prepared slides and navigate between slides. Navigation and annotation are captured in a log so that after class the student notes could be processed and integrated with the audio track for the class and presented on the Web for student reviewing.

On the right of Figure 1 is the interface provided for viewing audio-enhanced Web notes. The top frame shows thumbnail sketches of all slides from the lecture. The selected thumbnail image is magnified in the lower right frame. The lower left frame is divided into three main sections: a keywords sections shows words associated with the file to facilitate a content-based search; an audio section lists automatically-generated audio links indicating times in the lecture when that slide was visited; and a search link provides access to a search form for simple keyword search across all lecture notes. When an audio link is selected, an audio client is launched and begins playing the recorded lecture from that point in the lecture. We built our own streaming, indexable audio server and client players for this purpose. This audio service worked very well, but it was not cross-platform and students without access to a UNIX workstation were unable to take advantage of it.

We used this Classroom 2000 prototype during a 10-week graduate course and the results of our evaluation are reported elsewhere [1, 2]. This initial evaluation was much more of a feasibility experiment to see if we could get the technology to work for an entire course. We have gathered a lot of qualitative and quantitative data on this particular Classroom 2000 system, and analysis of that data has helped us to improve many aspects of the overall system.

We learned an important lesson about ubiquitous access in education. The lack of cross-platform audio prevented us from drawing any conclusions about how the availability of the Web notes influenced student and teacher strategies in the classroom. Not enough students had easy access to platforms that would run our homegrown audio services. We have since obtained a third party cross-platform audio solution (RealAudio) and are now in a position to better judge the impact of audio on note-taking practices.

Open research issues

Apart from the educational impact of the ubiquitous computing technology used in Classroom 2000, we can see some issues of more general concern to the CIA problem.

Granularity of integration In the shown in Figure 1, we were only able to provide audio links at the level of granularity of an individual slide. Every time a slide was visited by the student (or teacher) the log file was updated and an audio link with the associated time would appear in the Web notes. Other CIA systems provide a finer level of granularity, ultimately allowing any gesture (pen-down to pen-up) to produce an index into the audio or video stream. The reason we did not provide such functionality was technical (the Visual Basic ink object would not provide the information necessary), but it is not clear that the fine-grained indexing is optimal. Typically, the sound most closely relating to some annotation preceds the timestamp of the annotation itself, so there needs to be some intelligence in generating the link between annotation and audio. In our homegrown audio server, we took a first approximation at this intelligence by detecting the first pause in the audio track prior to the index point and beginning the audio stream from there. This was to simulate the beginning of a phrase or sentence by the speaker. Other researchers do not directly address how effective the fine-grained indexing was for the users during the access phase, though one report does note conventions in note-taking adopted to allow the user to be reminded how to adjust the system's timestamp to a more relevant audio index [15]. There will also be links generated that are not time related, one example of which we address next.

Supporting revision during access In our prototype, the Web notes were static, meaning that no additional information could be added to them after class. Upon reflection, this was obviously a bad choice, since it is common for some students to revise and rewrite their notes. In fact, the whole purpose of the audio-enhanced notes is to support this revision. Our later prototypes now support the ability to modify notes during review.

This additional capability introduces a subtle problem related to the granularity. How should notes referring to teh past be integrated with streams captured in the past? This problem already appears during the classroom or meeting session, especially when each gesture provides a time-index into audio. If a student goes back to some notes during class to modify them (perhaps adding some more words for clarity), should a new timestamp be generated that now points to a different part of the lecture, a part that may in no way be relevant to the revised notes? When modifying notes after class, what if the student simply wants to rephrase some of their notes but retain the link to the audio from the original gesture? In this case, static slide-level audio links might be preferred, or assigning timestamps to gestures that approximate the nearest neighbors. There are certainly some strategies that can be adopted, but how is this done in a way that predicts but does not interfere with the intention of the user? That is a very difficult question, complicated by our lack of experience as users of audio-enhanced notes.

Supporting networked interaction In a multi-user situation such as the classroom, there are many people taking notes. If we network all of the note-taking units, then a lot of interaction modes are possible, but which are useful? For example, we could allow students to "copy" the teacher's notes. Our analysis of student notes showed that a large number of them merely copied on their slides what was written by the teacher. This makes sense in a situation where the teachers notes are completely lost after the class. An electronic whiteboard solves that persistence problem, and some students (but not all) explicitly requested this copying service to allow them to pay more attention in class.

Students in the class also suggested the possibility of providing anonymous feedback during the lecture, to indicate whether the pace of the lecture was satisfactory or whether the material was boring or interesting, or to post questions that they might not otherwise ask. Realtime feeback during a lecture, if used properly, could actually improve a class. Even disregarding the realtime feedback, logging student reactions to lecture pace and content can be used to modify lectures later on, providing a service to the teacher that would otherwise be difficult to obtain.

CONTEXT-AWARE COMPUTING

Defining the problem

Future computing environments promise to free the user from the constraints of stationary desktop computing, so researchers should focus on what applications maximally benefit from mobility. Highly portable devices, such as personal digital assistants, pagers, and cellular telephones are starting to proliferate. The applications that the PDAs provide, however, are simple duplications of what we have on our desktops, and the other devices are still only used for simple messaging tasks. None of these devices take into account the one thing that changes most when a user is mobile —position. Building applications that are customized to the user's physical and even emotional context can be of benefit in stationary modes of interaction as well. Recall the example in Classroom 2000 where students would providing anonymous feedback on their reactions to lecture pace and content. A context aware Classroom 2000 system could signal the lecturer in some way if the majority of the students were either bored or feeling left behind.

The majority of context-aware is restricted to location-aware computing for mobile applications, and our own work started out that way. In thinking about and developing our own location-aware application, we were greatly influenced by work such as the PARCTab at Xerox PARC [20], the InfoPad project at Berkeley [13], the Olivetti Active Badge system [20] and the Personal Shopping Assistant proposed at AT&T [3]. A more general programming framework for describing location-aware objects was the subject of Schilit's thesis and reflected a lot of the work done at PARC [18].

Our experience: Cyberguide

There are a number of PDAs available today, yet there are not many people who are devoted to using them, for a wide variety of reasons. One way to thwart this lack of market success is to invent new ways to use it. We thought that the size of current PDAs was similar in size to a guidebook that a tourist might take with them on vacation. The book lists places to visit and provides important practical information, such as locations of interesting sights, or categorization of hotels and restaurants. The one thing a book does not know, however, is where the tourist is located when they want information. Position information augmenting an electronic guidebook would solve that problem, so we initiated the Cyberguide project to develop prototypes of a mobile, postiion-aware tour guide.

We further refined the scope of the project and decided to produce a prototype to support indoor tours of the GVU Lab during monthly open houses. These initial prototypes were reported on earlier [11] and were built using the original Apple MessagePad[2] and a very low-

[2]MessagePad is a registered trademark of Apple Computer, Inc.

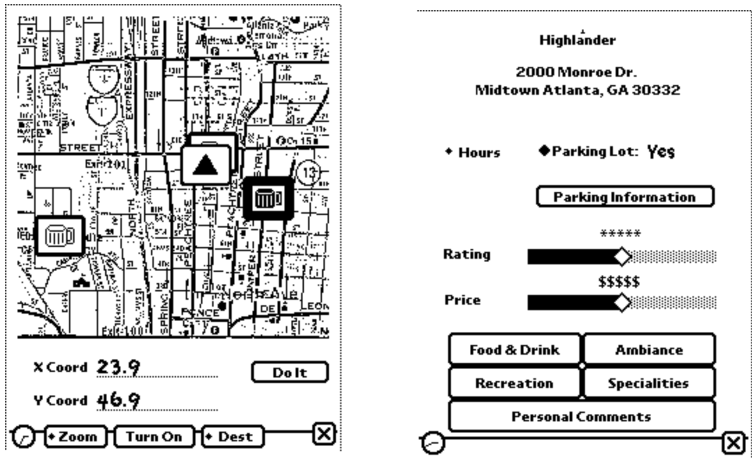

Figure 2: The CyBARguide interface. The left shows the interactive map indicating the user's location (the triangle) and the location of establishments previously visited (the beer mugs). The user modifiable database shown on the right supports the long-term development of touring information for a location.

cost, but effective cellular positioning system, consisting of an array of TV remote controls hanging from the ceiling of the lab. Each remote beamed a unique cell identifier to a transceiver unit connected to the MessagePad that was commonly used in robotics vision applications. We have since created several outdoor versions using GPS receivers to provide position information and have built an inexpensive indoor IR network to experiment with 2-way communication between the Cyberguide unit and the environment [12].

Figure 2 shows a version of the outdoor Cyberguide used for touring local establishments in Altanta.[3] As the user moves around, her location (the arrowhead in the picture on the left in Figure 2 is updated on the map. Interesting sights are indicated by a beer mug. Selecting a sight reveals an information sheet containing user-modifiable fields for that establishment. As the user tours around midtown Atlanta, she can add sights to the map and fill in whatever information she has on the place, useful for excursions on a later date.

Open research issues

Building several versions of the context-aware tour guide has generated a number of issues to do with the need for technology and as well as the research problem of context-aware mobile applications.

Providing ubiquitous positioning and communication In the design of our prototypes, we modularized communications capabilities and positioning to enable us to experiment with different realizations of each. Initially, we ignored communications capabilitties, but experimented with indoor and outdoor positioning systems. The indoor systems were cell-based, similar to the PARCTab and Active Badge systems, whereas the outdoor systems used the continuous model of GPS. Ultimately, we are aiming for a positioning service that works both indoors and outdoors. Unfortunately, GPS is not uniformly available indoors, and it is entirely impractical to use the current IR-based indoor solution outdoors. To get the resolution of GPS uniformly across all space will require a hybrid solution.

Communication is important for several reasons. It is required in order to keep track of the location of all mobile units and to broadcast information to them. It can provide the ability for a mobile tourist to issue requests, and we experimented with this to allow visitors to GVU the ability to send e-mail expressing interest in a project being demonstrated and have it delivered directly to a contact person for that project. Also, our view of the mobile unit was not one of a mass storage device, so it will soon become necessary for information (a map of a location or a description of an interesting sight) to be delivered on demand to the unit, rather than stored locally. Even if the mobile unit could store all of the information a tourist would need, it would be static. Current automobile on-board navigation systems use a static database of local streets. Such a system is not very useful if you are interested in knowing which roads to avoid because of accidents or construction. Despite this apparent need for communications, current commercial units are very communications poor.

There is an interesting relationship between the positioning and communication systems. As we already noted, for applications in which you want objects to know about the position of other objects, there must be some sort of communication, as shown done in the PARCTab and Active Badge systems. However, it can be impossible or undesirable to couple positioning and communication together. For example, if position is coming from GPS, then a separate means of commu-

[3] This version of Cyberguide bears the affectionate moniker CyBARguide.

nication must be used. In our current version of indoor IR positioning using the Sharp IR units, we can couple positioning and communication, but the range of the IR link is so limited (3 feet) that communication is cumbersome. Whereas it makes sense to use a short range IR positioning system that doesn't cover all space but where it is available provides more exact positioning (and orientation) information, communication needs to be uniform throughout some space.

There is more to context than position We currently have a very limited notion of context in Cyberguide —physical location and crude orientation. We have experimented with capturing historical context (what sights have been visited already), but there are a number of other aspects of the tourist's context that can be useful. For instance, knowing where everyone else is located might suggest places of potential interest. Knowing the tourist's reaction to some exhibits would help in suggesting other related places of interest. Being aware of time of day or day of week may aid in more intelligent user queries of interesting attractions or activities.

Context awareness should be done behind the scenes. The more that can be automatically captured and turned into context, the better. If the user has to explicitly inform the system about context information ("I am currently located here." or "This exhibit is boring to me." or "The museum is currently closed."), she probably will not bother to do so. This insight also extends to our work in providing anonymous feedback in Classroom 2000. The more a student has to explicitly comment on pace or level of interest, the less likely they will do so.

Use of personalized vision and voice technology We have been using Cyberguide at GVU open houses for over a year and one discouraging note is that even though users quickly understand how the system works, they tend to stand in one place playing around with the new toy and don't use it to guide exploration. The interface is getting in the way of the task! There are technical solutions that might help this problem. The hand-held map could be replaced with a heads-up display, using a personalized vision system in an augmented reality mode of interaction. Or we can simply use voice technology coupled with position to provide a more useful directory information service. A more important factor to consider is how a tourist wants to interact with the guidebook. We defaulted to presenting a map of the area of exploration, but that is sometimes not an appropriate initial presentation. We also need to experiment with larger areas to determine if the effect we saw was influenced by the relatively small (2500 square feet) GVU lab space that was being toured.

UBIQUITOUS SOFTWARE SERVICES

Defining the problem

The desktop paradigm leaves it to the user to find the interface to a computational service (such as an e-mail browser or a calendar manager) when it is needed. A ubiquitous software service, on the other hand, finds the user. Two important characteristics of such a service are: its availability on any device handy to the user; and its adaptability to a changing set of services that the user wants. The former characteristic is referred to as the scaleable interface problem. The creation of an architecture-neutral virtual machine, such as the Java Virtual Machine, solves part of the scaleable interface problem because it now becomes possible to execute the same program on many different devices.

The latter problem, one of service integration, can be solved with a large suite of services that are tightly integrated and provide an overall service greater than the sum of the individual services. This is the solution seen in commercial personal productivity packages, and that is to provide. Tight integration is not the best solution because it requires much effort on the part of the designer of a single service. It also limits the flexibility of the end user, since the integration of tools is limited to those in the suite.

Our experience: CyberDesk and CyberNag

Elsewhere in these proceedings, we describe our work on the CyberDesk project, the main goal of which was to provide a Java-based integration framework for allowing network services and personal productivity tools to work together without additional programming effort by the designer of the service or its user [25]. CyberDesk knows what services are available to the user and what types of data a service displays and accepts. When the user highlights any information displayed by a service, CyberDesk automatically suggest actions that can be invoked on other services using the highlighted information. The burden for integrating services is thus removed from the user, as CyberDesk causes the behavior of a service to find the user when relevant.

As another example of this research theme, we considered the domain of personal communication. Today we have many ways to send messages to someone. We can phone, fax, e-mail, or page someone in addition to writing a traditional letter. Unfortunately, the limitation that comes with this variety of modes of communication is that the input mode chosen by the sender dictates the output mode the receiver must use to accept the message. A ubiquitous messaging service should remove this dependency, being able to locate a message recipient, determine what mode of communication is most suitable to reach her, and deliver the message, suitably transformed to the output mode.

We created the CyberNag system to provide this ubiquitous messaging service and instantiated it with the capability to accept e-mail and deliver it as e-mail, page or phone call. When e-mail is sent to the recipient, the CyberNag system determines if e-mail is an appropriate delivery mode, based on urgency of the message and location of the recipient. If the system decides to deliver the message as a page, it is summarized and sent out as an alphanumeric page. If a phone call is the choice, the system initiates a phone call to the user and allows the option to hear the message read to them.

Open research issues

We have relatively little experience evaluating the use of CyberDesk and CyberNag, since they are some of our most recent projects. We can, however, offer the following insights into problems that remain unsolved.

Scaleable interfaces We have not really addressed the problem of scaleable interfaces yet. CyberDesk assumes that everything will eventually be available as a Java applet within a Web browser. We are only now seeing the availability of Web browsers and the Java virtual machine on anything other than a traditional personal computer, so the issue of providing similar functionality on radically different interfaces will soon emerge. Since Web browsing is the current killer app, and it transfers nicely to a task someone might want to do on a number of devices, we should soon start to see results for scaleable browsing. Assuming we have reasonable wireless connectivity, scaleable services such as calendar management will also follow. One of the biggest problems with the adoption of PDAs is that they become isolated applications, when in fact most users have personal information in some sort of digital format that they would like to easily and continually access from their desktop. Attempts to integrate portable devices with the desktop, such as the US Robotics Pilot or the Dynamic Interface Libraries Apple provides for coordinating Newton OS applications with a personal computer, are a start, but they still rely on separate programming of applications to run on the different platforms. What happens when the user wants to use a phone to access the calendar?

Ubiquity should not be annoying One obvious social problem with ubiquity is the fear that the user may never be able to hide from the interface. It is an obvious concern for CyberNag, so there has to be consideration for allowing the user to ignore messages. Despite the suggestion of pestering that comes with the name, CyberNag is really intended to weed out unimportant messages and only deliver the most urgent ones.

The problems of ubiquity arise in CyberDesk as well. As a user acquires more and more services, the opportunities for detecting potential integration between any two dramatically increases, up to the point where the user is overwhelmed by too many choices for what to do next. Work in this area needs to concentrate on solutions for revealing function only when it is most likely desired.

OTHER ISSUES WITH UBIQUITOUS COMPUTING

We have defined three major themes and open issues for ubiquitous computing research that have come from our own experience building applications that rely on ubiquitous computing technology. We also have some general insights to share on our applications-centered research method.

Cross-pollination of themes

Though we presented each project within the boundaries of a discussion on one research theme, a single project will demonstrate more than one of these research themes. For example, the user-modifiable database in CyBARguide suggests the usefulness of capturing a tourist's experience in order to facilitate access to that information later. Providing anonymous feedback in Classroom 2000 suggests that possible usefulness of context-awareness. CyberNag provides a ubiquitous messaging service, but the bulk of its utility comes from awareness of the context of the message recipient, knowing where she is located and what interfaces are most accessible to her at a given time.

The advantages and disadvantages of keeping cost down

The FCE Group does not currently have the luxury of a heavy financial endowment, so have been forced to keep costs down.[4] The obvious disadvantage is that we have often been forced to work with unreliable and unsupported hardware only because we could obtain it relatively cheaply through resellers. There are two distinct advantages, however, to frugal ubiquitous computing. First, because we found inexpensive hardware (mainly pen-based units) we could buy more of them. This was essential for Classroom 2000 because we are trying to investigate the impact when everyone had their own electronic notebook. It is also important for our applications work with hand-held units because we are trying to portray them more as community assets and not personal devices. Second, the more units we purchased, the more independent prototypes we could produce. Ideas realized as prototypes were the single best source of new ideas, both from ourselves and from others who saw demonstrations.

Engineering, technology push and HCI research

Our research approach has been very much a technology push, but there is no need to apologize for that. Humans have often demonstrated their inability to predict the impact of technology in everyday life. Our rapid prototyping of novel applications of ubiquitous technology is simply an admission that we do not know the best way to use what is quickly becoming available to make our lives better. We understand now what Alan Kay meant when he said that the best way to predict the future is to invent it. We adapt this statement slightly to add that the best way to invent is to experiment as much as possible. And experimentation often boils down to an engineering problem.

To evaluate the affect of the ideas in Classroom 2000, for example, first requires a system that will reliably function on a daily basis. This engineering is a necessary evil, but it only remains evil if no evaluation ever occurs.

CONCLUSION

We have defined three general research themes for ubiquitous computing that have emerged from our own experience and applications perspective. These thems are: automated capture, integration and access; context-awareness; and ubiquitous software services. We have at-

[4]In fact, we were tempted to entitle this article "Ubiquitous computing for five dollars a day, in the spirit of Pausch's similarly titled piece for virtual reality [16].

tempted to make the abstract themes concrete in terms of systems we have built that portray important features and highlight a number of open issues. Many of these issues are targets for our future research, but we hope that they might interest other HCI researchers who concur with our research method of experimental rapid prototyping and evaluation in real-life context.

ACKNOWLEDGMENTS

Will be included upon acceptance.

REFERENCES

1. G. D. Abowd, C. Atkeson, A. Feinstein, Y. Goolamabbas, C. Hmelo, S. Register, and N. N. S. and-Mikiya Tani. Classroom 2000: Enhancing classroom interaction and review. Technical Report GIT–96–21, GVU Center, Georgia Institute of Technology, September 1996.
2. G. D. Abowd, C. G. Atkeson, A. Feinstein, C. Hmelo, R. Kooper, S. Long, N. Sawhney, and M. Tan. Teaching and learning as multimedia authoring: The classroom 2000 project. In *Proceedings of Multimedia'96*, 1996. To appear.
3. A. Asthana, M. Cravatts, and P. Krzyzanouski. An indoor wireless system for personalized shopping assistance. In L.-F. Cabrera and M. Sattyanarayanan, editors, *Workshop on Mobile Computing Systems and Applications*, pages 69–74. IEEE Computer Society Press, December 1994.
4. L. Cabrera and M. Satyanarayanan, editors. *Proceedings of the Workshop on Mobile Computing Systems and Applications.* IEEE Computer Society Press, 1995.
5. L. Degen, R. Mander, and G. Salomon. Working with audio: Integrating personal tape recorders and desktop computers. In *Proceedings of CHI'92*, pages 413–418, May 1992.
6. A. Dix, J. Finlay, G. Abowd, and R. Beale. *Human–Computer Interaction.* Prentice Hall International, 1993.
7. S. Elrod, R. Bruce, R. Gold, D. Goldberg, F. Halasz, W. Janssen, D. Lee, K. McCall, E. Pedersen, K. Pier, J. Tang, and B. Welch. Liveboard: A large interactive display supporting group meetings, presentations and remote collaboration. In *Proceedings of CHI'92*, pages 599–607, May 1992.
8. D. Hindus and C. Schmandt. Ubiquitous audio: Capturing spontaneous collaboration. In *Proceedings of CSCW'92*, pages 210–217, 1992.
9. T. Imielinski and H. F. Korth, editors. *Mobile Computing.* Kluwer Academic Publishers, 1996.
10. T. K. Landauer. *The Trouble with Computers.* A Bradford Book. MIT Press, 1995.
11. S. Long, D. Aust, G. D. Abowd, and C. G. Atkeson. Rapid prototyping of mobile context-aware applications: The cyberguide case study. In *Proceedings of CHI'96*, 1996. Short paper.
12. S. Long, R. Kooper, G. D. Abowd, and C. G. Atkeson. Rapid prototyping of mobile context-aware applications: The cyberguide case study. In *Proceedings of MobiCom'96*, November 1996. To appear.
13. A. C. Long, Jr., S. Narayanaswamy, A. Burstein, R. Han, K. Lutz, B. Richards, S. Sheng, R. W. Brodersen, and J. Rabaey. A prototype user interface for a mobile multimedia terminal. In *Proceedings of CHI'95*, 1995. Interactive experience demonstration.
14. S. Minneman, S. Harrison, B. Janseen, G. Kurtenbach, T. Moran, I. Smith, and B. van Melle. A confederation of tools for capturing and accessing collaborative activity. In *Proceedings of Multimedia'95*, November 1995.
15. T. Moran, P. Chiu, S. Harrison, G. Kurtenbach, S. Minneman, and W. van Melle. Evolutionary engagement in an ongoing collaborative work process: A case study. In *Proceedings of CSCW'96*, 1996.
16. R. Pausch. Virtual reality on five dollars a day. In *Proceedings of CHI'91*, pages 265–270, 1996.
17. H. Rheingold. *Tools for Thought.* Prentice Hall, Englewood Cliffs, New Jersey, 1985.
18. W. N. Schilit. *System architecture for context-aware mobile computing.* PhD thesis, Columbia University, 1995.
19. L. J. Stifelman. Augmenting real-world objects: A paper-based audio notebook. In *Proceedings of CHI'96*, pages 199–200, April 1996. Short paper.
20. R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992.
21. K. Weber and A. Poon. A tool for real-time video logging. In *Proceedings of CHI'94*, pages 58–64, April 1994.
22. M. Weiser. The computer of the 21st century. *Scientific American*, 265(3):66–75, September 1991.
23. M. Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):75–84, July 1993.
24. S. Whittaker, P. Hyland, and M. Wiley. Filochat: Handwritten notes provide access to recorded conversations. In *Proceedings of CHI'94*, pages 271–277, April 1994.
25. A. Wood, A. Dey, and G. D. Abowd. Cyberdesk: Automated integration of desktop and network services. In *Proceedings of CHI'97*, 1996. Technical note submitted for review.