

Classroom 2000: An experiment with the instrumentation of a living educational environment

by G. D. Abowd

One potentially useful feature of future computing environments will be the ability to capture the live experiences of the occupants and to provide that record to users for later access and review. Over the last three years, a group at the Georgia Institute of Technology has designed and extensively used a particular instrumented environment: a classroom that captures the traditional lecture experience. This paper describes the history of the Classroom 2000 project and provides results of extended evaluations of the effect of automated capture on the teaching and learning experience. There are many important lessons to take away from this long-term, large-scale experiment with a living, ubiquitous computing environment. The environment should address issues of scale and extensibility, it should continuously be evaluated for effectiveness, and the ways in which the environment both improves and hinders the activity that it aims to support—in our case, education—need to be understood and acted upon. In describing our experiences and lessons learned, we hope to motivate other researchers to take more seriously the challenge of ubiquitous computing—the creation and exploration of the everyday use of computationally rich environments.

Future computing environments hold promise for providing valuable services through pervasive use of computation. One service could be the capture of everyday experiences of its occupants, making that record available for later use. We can spend much time listening to and recording, more or less

accurately, the events that surround us, only to have one important piece of information elude us when we most need it. We can view many of the interactive experiences of our lives as generators of rich multimedia content. A general challenge in *ubiquitous computing* is to provide automated tools that support the capture, integration, and access of this multimedia record.^{1,2} Automated support can help computers do what they do best—record an event—in order to free humans to do what they do best: attend to, synthesize, and understand what is happening around them, with full confidence that specific details will be available for later perusal. Automated capture can be viewed, therefore, as a paradigm for multimedia authoring.

In order to instrument any environment to automate multimedia content generation we need to provide computational services in that environment that are effectively pervasive without being overly intrusive. This is a goal in common with much of the work over the past decade in the area of ubiquitous computing.¹ There are many difficult issues in constructing and operating such a ubiquitous computing environment,²⁻⁴ but we agree with Weiser's³ sentiment that

©Copyright 1999 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

the applications are the whole purpose for doing ubiquitous computing research. In this spirit, we describe the Classroom 2000 project, which presents an educational application of the automated capture problem in ubiquitous computing.

The Classroom 2000 project is an attempt to support both teaching and learning in a university through the introduction of automated support for lecture capture. Whereas most work in courseware development focuses much energy on the development of multimedia-enhanced materials, Classroom 2000 attempts to finesse the issue of content generation. By instrumenting a room with the capabilities to record a lecture, we are trying to make it easy to produce multimedia courseware. Our goal in the project is twofold:

- We want to understand the issues in designing a ubiquitous computing application that provides effective capture and access capabilities for rich live experiences.
- We want to understand what it takes to produce a robust, ubiquitous computing application whose impact in its targeted domain can be evaluated over a long period of time.

Over the three-year lifetime of the project, we have gained extensive experience in the use of an instrumented environment to support automated capture. One of the main goals of the project was to build an instrumented classroom environment in which capture was made reliable and easy. Only after that reliability and ease of use was achieved, we postulated, would we be able to assess the value of the system as a learning and teaching aid.

We have produced a system that has captured parts or all of over 60 courses at Georgia Institute of Technology. Courses have ranged from graduate to undergraduate and have been in areas of computer science, electrical engineering, and mathematics. An installation at Kennesaw State University in northwest Atlanta has been in operation for just under a year and has captured lectures in nine courses in the College of Science and Mathematics. New installations are planned for use in 1999 for two more classrooms in the College of Computing and up to seven classrooms in the School of Electrical and Computer Engineering at Georgia Tech. A new classroom installation is planned for use at Georgia State University, in downtown Atlanta, starting in the fall 1999 quarter.

Other universities around the country have made plans to install versions of the system for research and educational purposes over the next year. This continued interest and use of the system both locally and elsewhere is one clear measure of success. As discussed in this paper, one of the keys to a successful ubiquitous computing research project is that there is a compelling application of the technology. Attempting to provide support for classroom lecture capture is clearly a compelling application of automated capture.

This paper provides a historical account of the Classroom 2000 project, divided into the three main stages of the project. The first stage covers the early feasibility experiments and evaluation results from single-class experiments with our initial prototype capture system. The second stage of the project was ushered in by the construction of a special-purpose instrumented classroom—our so-called *living laboratory*—which opened in January 1997. It includes a more mature capture system used to support many simultaneous courses, allowing more in-depth evaluation of the effect of capture on teaching and learning practices. The third and current stage of the project extends the capabilities of the system.

Our experiences with Classroom 2000 have given us a number of insights into the problem of automated capture as well as human-computer interaction (HCI) and software engineering issues for ubiquitous computing. This paper describes some of those insights, summarizes important related work, and discusses future goals for this project.

Stage 1: Feasibility studies

In July of 1995, our research began with the goal of producing a classroom environment in which electronic notes taken by students and teachers could be preserved and augmented with audio and video recordings. The initial idea was to produce media-enhanced records of a traditional lecture. The project philosophy followed a number of explicit strategies:

- We named the project Classroom 2000 to indicate that, while we were eager to build some infrastructure that was not available in any other university classrooms we knew of, we also intended that our solutions would be realizable in any university within five years.
- Our overriding goal was to be able to capture enough of the lecture experience to provide stu-

dents with both short- and long-term benefits. However, we were not sure what would be the best use of this capture capability, so we wanted to quickly get to the point where the instrumentation and electronic augmentation of the classroom were no longer the focus of attention. The novelty had to wear off so that we could begin to observe what real effect the capture capabilities had on teaching and learning styles.

- Although we knew that automated capture would have applications in distance learning and general business meetings, we explicitly focused on the standard, synchronous university lecture. Although there are some who would argue the effectiveness of this age-old didactic form, the fact remains that a vast majority of education occurs this way. Producing a system specifically tuned to the traditional lecturing style would allow us to experiment with a large number of users. It would also put us in a position to observe how automated capture affects the form of the traditional lecture.

With these strategies in mind, this paper provides a historical account of how the project has evolved since July 1995.

The initial prototype. The first six months of the project consisted of a number of brainstorming sessions with interested faculty and students, attempting to create a vision of what automated capture would mean in the classroom. We operated with a hard deadline of six months to develop a prototype to be used in one class for an entire 10-week quarter.

We knew from the onset that the prototype systems would be subject to many revisions due to changes in hardware and in requirements that we could not predict. Very early on, we settled on an architectural scheme that would allow the system to evolve over time. This scheme remains today in our structuring of the system and is described in four phases:⁵

1. **Preproduction.** We make an analogy to the production stages of the film industry. First are the activities in preparation of the live event. For a lecture, this includes activities that the teacher performs in order to prepare for the lecture. We made a conscious decision to minimize the amount of extra work the lecturer had to do to prepare for a lecture that would be captured using our system.
2. **Live capture.** Once a lecture begins, we attempt to capture, with a time stamp, all of the relevant

activity. We understand the live experience as a generator of various *streams* of information. For example, the set of prepared slides that might be shown during the lecture constitutes one stream that is divided into discrete chunks, or slides, that appear at various times. Handwritten notes, made either by the teacher or a student, constitute another stream, with individual strokes (pen-down to pen-up) defining the time-stamped chunks. Other examples of streams include the audio or video recorded by one or more microphones or cameras, a series of URLs (uniform resource locators) visited with a Web browser during a recorded session, or even a series of user interactions with a program that is being demonstrated during the lecture. This phase ends when the live session, such as the class lecture, ends.

3. **Postproduction.** The whole purpose of defining and capturing streams is to support the development of interfaces that integrate or associate related streams. For example, we wanted to be able to look at any part of a lecture after class and associate it with what was being said during that part of the lecture. This integration activity is central to the postproduction activity. All of the captured streams are made available for the generation of a variety of interfaces that students will access in the next phase.
4. **Access.** The final phase comes later, when the students access some record of the live experience. It is important to realize that in this phase, the physical interfaces used by the students will be very different from the physical interfaces used during live capture. For example, very few students would be expected to possess a networked, pen-based computer, so we cannot assume that form of interaction will be available. Since access needs to be universal and across platforms, it is important to build on an existing infrastructure, such as the Web.

We initially supported one instructor (the author) in teaching an introductory graduate course in human-computer interaction. Lectures were given as prepared presentations of 10–20 slides for a 90-minute lecture. The slides were shown on a Live-works, Inc., LiveBoard^{**}, a 67-inch diagonal upright pen-based computer,⁶ and the lecturer navigated between slides and wrote on them as the lecture progressed. The lecturer interface is shown in Figure 1 and was a Visual Basic^{**} program we wrote, called “ClassPad.” Some students in the class were given

hand-held tablet computers that ran the same ClassPad program. These students could follow along in the lecture and write whatever they wanted on their units. Simplicity was important in the design of the note-taking interface of ClassPad. We wanted to maximize the space used for presenting slides, and for writing by both teacher and students.

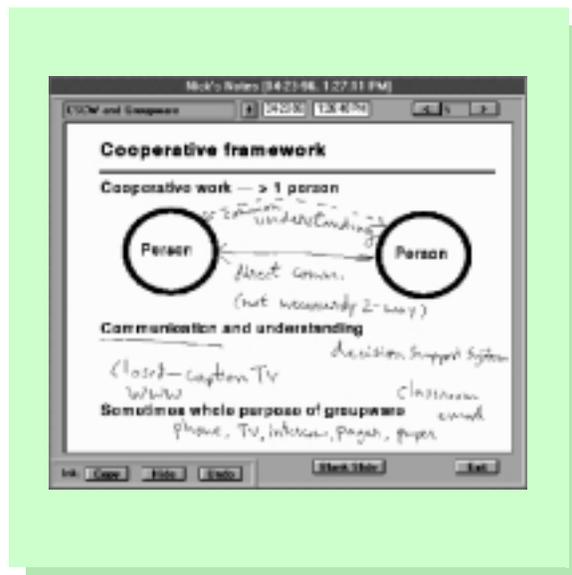
ClassPad was used very much like a slide presentation tool, such as Microsoft PowerPoint**. Simple navigation buttons (arrows in the upper right corner) allow the user to go forward and backward in the presentation, one slide at a time. The pen can be used to write on the slide as the lecture progresses. As needed, a new blank slide can be inserted and written upon. Every time a slide was visited, ClassPad would record the time.

After class, the lecturer and student slides were converted into a series of HTML (HyperText Markup Language) pages (see Figure 2). This interface was generated by a series of Perl scripts operating on the annotated slides downloaded from the LiveBoard and student units. The top frame provided thumbnail images of all slides for that lecture. The user selected one thumbnail image, and the full-sized image appeared in the lower-right frame. The lower-left frame contains a list of keywords associated with the slide, the automatically generated audio links representing each time the slide was visited during the lecture, and a link to a form that allows keyword searching across all slides for the entire course. The lecture was recorded in a digital audio format, so that clicking on an audio link launched our own external streaming audio player at that point in the lecture.

None of the machines used in this initial prototype was networked, which meant there was a very high overhead both before and after lectures to upload and download materials. This overhead was estimated at four hours per lecture, and this was a serious roadblock to expanding our efforts.

Results from initial evaluation. The purpose of this first experiment was not to obtain much significant evaluation data, but to experiment to see what was required to support a class with this kind of capture for an entire quarter, and to prove that we could do it. However, we asked students to keep a journal throughout the quarter and to fill out a simple questionnaire at the end. These results and other observations are summarized here. More details can be found elsewhere.⁵

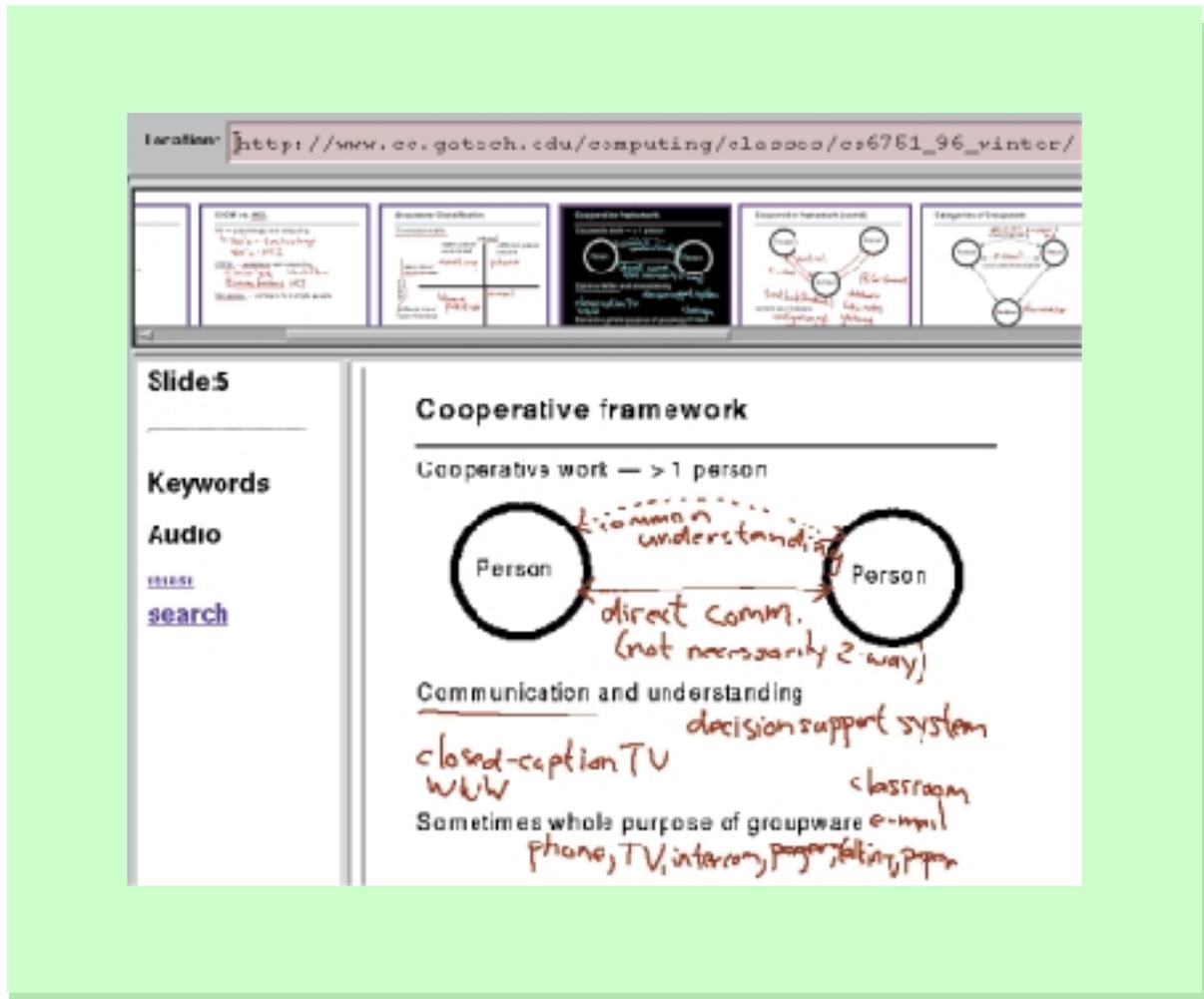
Figure 1 The initial ClassPad interface. This same interface was used by teachers and students.



In this first experiment, the teacher used the ClassPad system on a LiveBoard and some students used small tablet computers to take notes. We did not have enough student units for every student in the class. Four volunteers were selected to take notes for the remainder of the quarter. The rest of the units were made available on a first-come, first-served basis. The students can be categorized in three groups: the four volunteers who used the student units in every lecture, another four students who chose not to take notes electronically for any of the lectures, and the remaining 17 students in the class, who used the student units an average of 2.9 times each. There were two important observations concerning the use of student units and the overall reaction to the technology.

Students' electronic notes were often very similar to the lecturer's notes, as can be seen in Figure 3, even though the lecturer's notes were made available via the Web through the same interface. In addition, the units were slow and small and required more student attention than traditional paper and pen. Since research supports our general intuition that copious note taking is not a good learning practice, we viewed this as a bad sign.⁷ We noticed this trend in both quarters that we used student note-taking units (Winter and Fall 1996), and it was one of the reasons we decided temporarily to end support for student units.

Figure 2 Frame-based Web presentation of lecture notes with annotations and audio links



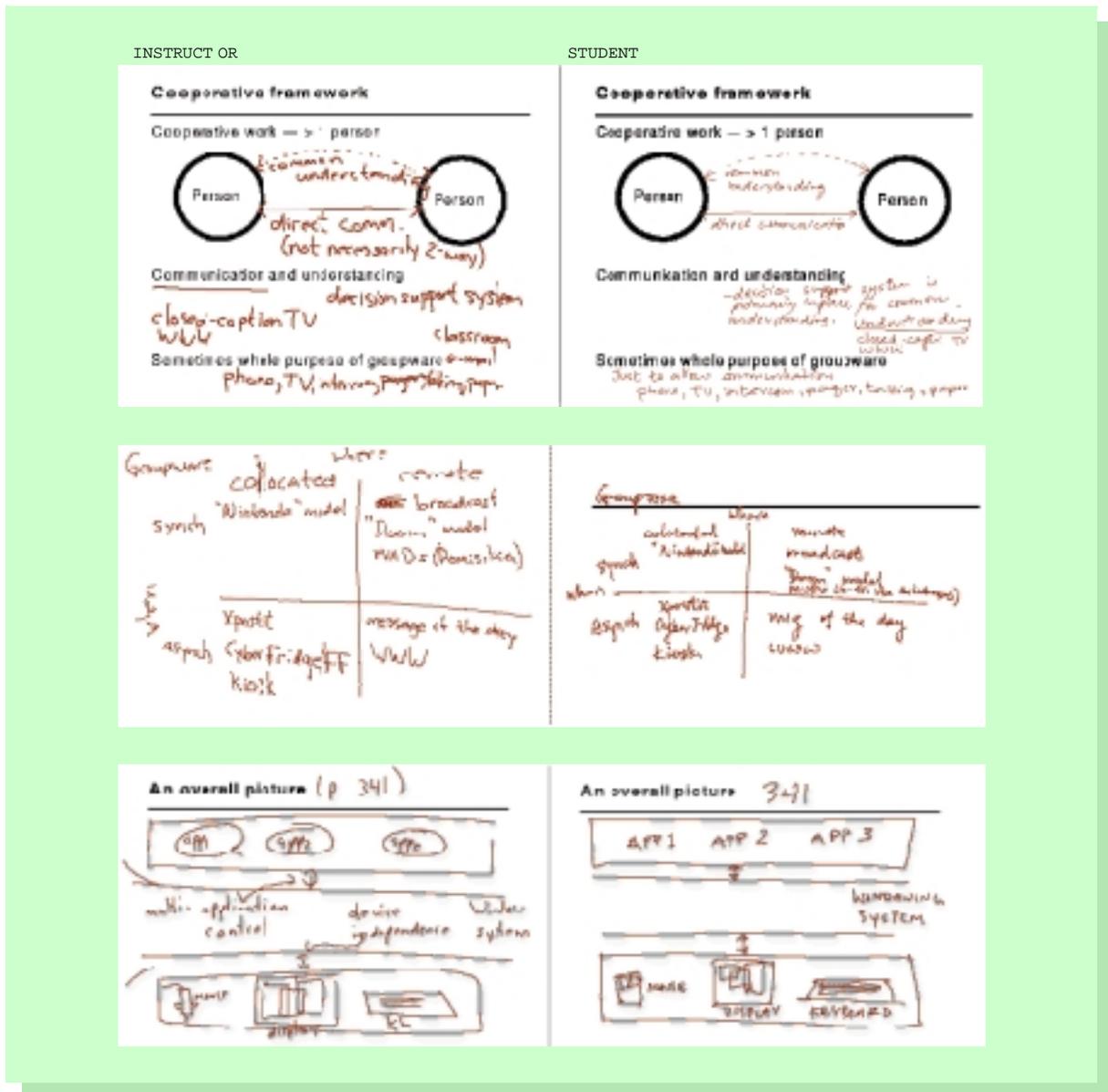
General reaction to the system was surprisingly positive. Of the 25 students, 24 filled out a questionnaire on their reactions to the use of the technology in the class. The objective questions asked about overall impressions and then specifically about the ClassPad note-taking application and the use of the LiveBoard with Web-based review notes. Possible answers were: 1 (strongly disagree), 2 (disagree), 3 (neutral), 4 (agree), and 5 (strongly agree). Table 1 summarizes the results of the objective portion of the questionnaire.

The strongest positive reaction was in how the prototype was perceived to contribute to learning the

particular subject matter, and this is not surprising. The course was on HCI and the students were themselves experiencing a new interface in the classroom. In addition, the project work was based on developing and evaluating ideas for new Classroom 2000 prototypes, and the students appreciated the authenticity of redesigning a system they were currently using.

One of the initial goals of Classroom 2000 was to examine the effect of personal interfaces in the classroom. Our initial observations show that the students were most negative toward the personal electronic notebooks. The LiveBoard and Web notes together

Figure 3 Comparison of teacher (left) and student (right) notes. Notice the high degree of similarity.



comprised the most desirable technology from the students' perspective. In fact, the most positive reactions to the overall system came from those students who never used the personal note-taker, and the least positive reaction came from those who used it for every lecture. This observation also caused us to reconsider the value of the personal units. One explanation for this result is that students who did

not have to labor with the note-taking devices derived all of the benefit of capture without incurring any additional cost. Students who used the note-taking devices derived the same benefit, but at the cost of using the cumbersome devices. Grudin talks about this phenomenon in groupware systems, in which it is important to understand who does the work and who gets the benefit.⁸

Table 1 Reaction of students to overall technology, electronic notebook, and LiveBoard with Web notes

Student Reaction	Topic (Number of Questions)	Average (σ)
Overall	Was desirable technology (11)	3.67 (.98)
	Was easy to use (2)	3.02 (1.23)
	Increased effectiveness of class (9)	3.62 (.99)
	Improved class participation (2)	3.40 (.88)
	Contributed to learning subject (2)	3.94 (.86)
Notebook	Was desirable technology (1)	3.13 (1.03)
	Was easy to use (3)	3.13 (1.14)
	Increased effectiveness of class (1)	2.88 (.90)
	Helped me take fewer notes (2)	2.85 (.87)
LiveBoard	Was desirable technology (2)	3.87 (.82)
	Was easy to use (3)	3.68 (1.09)
	Increased effectiveness of class (1)	3.29 (1.04)
	Helped me take fewer notes (2)	2.88 (1.00)

There was very little use of the audio augmentation provided by the Web-accessible notes. Upon further investigation, we understood this to be a result of the platform and network requirements that reliable audio service required. We had created our own streaming media system, but it was limited to use on only major UNIX** platforms. Few students had UNIX machines in places where they studied. To better evaluate whether audio augmentation was valuable, we needed to provide a more universally accessible service.

Stage 2: A living laboratory

In January of 1997, a new classroom, shown in Figure 4, was opened that was specially instrumented for Classroom 2000 use. Microphones and video cameras were embedded in the ceiling. A cabinet centralized all of the signals from the microphones and cameras and also provided backup units (DAT [digital audio tape] recorder and VCRs [video cassette recorders]). Audio and video signals were distributed out of the cabinet to a variety of digital encoding computers. A LiveBoard was again used as the electronic whiteboard. Two ceiling-mounted projectors attached to networked computers were also available, giving a total viewing area approximately the size of a traditional classroom whiteboard. The room seats 40 students, with electrical power and network access to each seat.

The main advantage of the room was that it provided all of the capabilities needed for capture in a per-

manent setup. We could now set our sights on creating a system that would allow the capture of many courses during the same quarter. In order to do this, there were several engineering issues that had to be addressed. We needed to streamline the integration of the various production stages (preproduction, live capture, postproduction, access). We also felt that to make lecturers more comfortable in using the electronic whiteboard, it had to look and feel like the traditional whiteboard.

We developed an electronic whiteboard system in Java**, called Zen★ (“zen-star”), that was operational by the Spring 1997 quarter. The Zen★ system was designed to support multiple, simultaneous captured sessions, in different locations. The main system consisted of a threaded central server, ZenMaster, which handled the requests from a number of clients running our ZenPad electronic whiteboard software. Figure 5 shows a screen shot of the original ZenPad client. As you can see, space for the whiteboard was maximized, and the interface provided only simple scrollbar navigation between prepared slides with the ability to annotate individual slides. Pull-down menus allowed the teacher to change pen color or line thickness.

After class, students could access the notes by invoking the same ZenPad system in a viewing mode. The default mode of ZenPad was determined by the user name upon log in. Students were only allowed to view lectures, while teachers were allowed to view and create lectures. When a user was viewing a lecture, as shown in Figure 5, selecting a single recorded gesture, such as the highlighted “0” in “60’s,” would launch the audio at the time when that gesture was written during the lecture. This was an improvement over the previous ink-audio integration of ClassPad, which was only possible at the level of slide visits.

The original Zen★ system was robust enough to support seven separate classes during its first quarter of use. The protocol between the client and the server had to be continually debugged throughout the quarter, but for the most part all lectures were captured for the quarter. Starting the lecture in live capture mode, as well as ending the lecture and moving all captured data to a central server, however, required much manual effort, a point that is discussed later.

Approximately two out of every ten lectures were not captured due to some system error, but teachers and students were beginning to rely on having the instrumented room take notes for them. This was

Figure 4 This prototype instrumented classroom opened in January 1997.



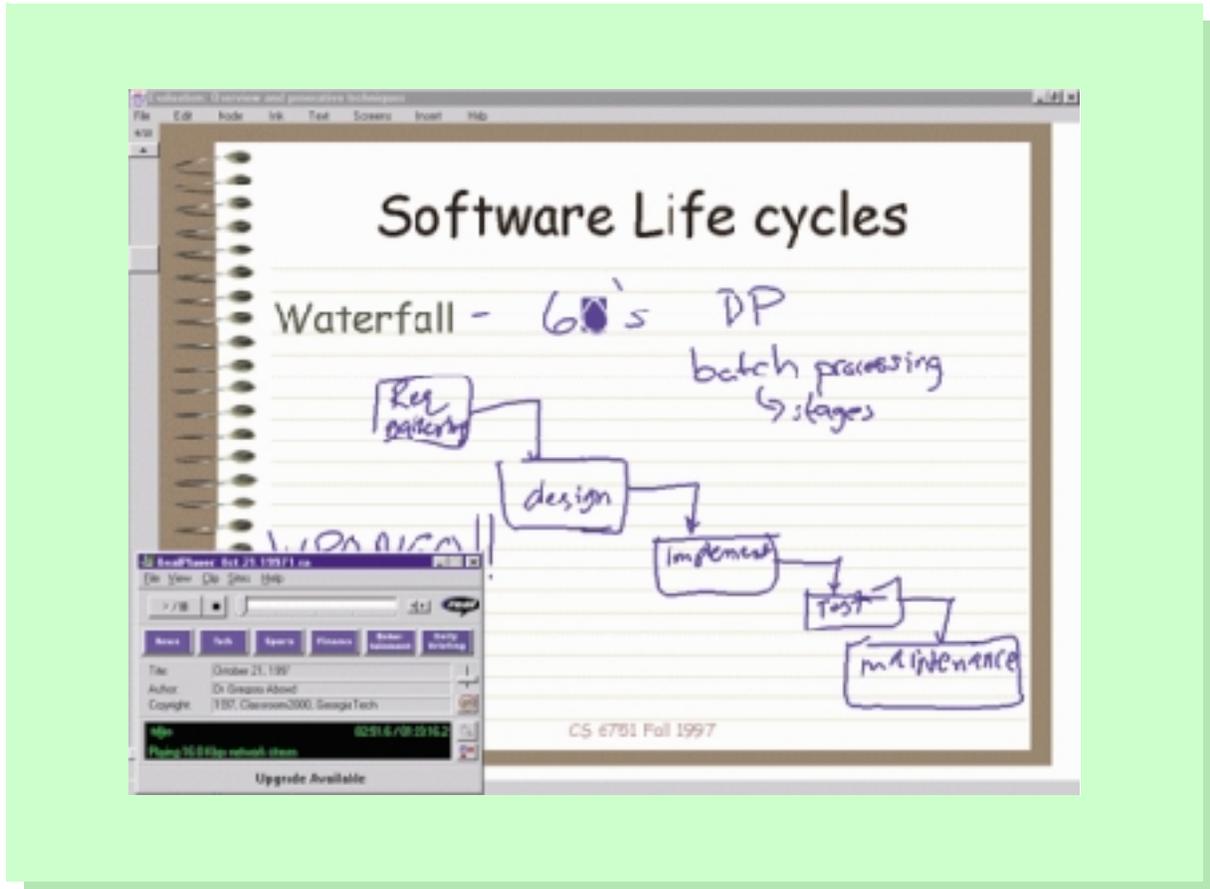
both a blessing and a curse. The blessing was that about 18 months into the project we were nearing the point at which we could begin to assess the value of automated support for capture from the user's perspective. We could not yet measure any learning benefit, but we could better understand how the system was used by its intended user population in everyday use. The project finally had a *living* laboratory for research, a critical element to any ubiquitous computing applications research.

The "curse" of our situation was that we now really needed to listen to and react to the requests of our users. While the ZenPad system was effective enough as a capture interface during lectures, it failed mis-

erably as an access interface. At the time of its introduction, there were very few reliable browsers that could support applets written in Java. The variety of machines that students were using required a much more cross-platform and robust access interface, or students would ignore the notes altogether.

One professor using the system that quarter felt particularly affected by this situation and pushed for a return to the HTML-only access interface of previous quarters (see Figure 2) that would reliably and quickly load on student machines. We could have reverted to using the older postproduction tools to produce those interfaces from our captured data, but those interfaces did not allow audio links at a finer

Figure 5 The original ZenPad interface

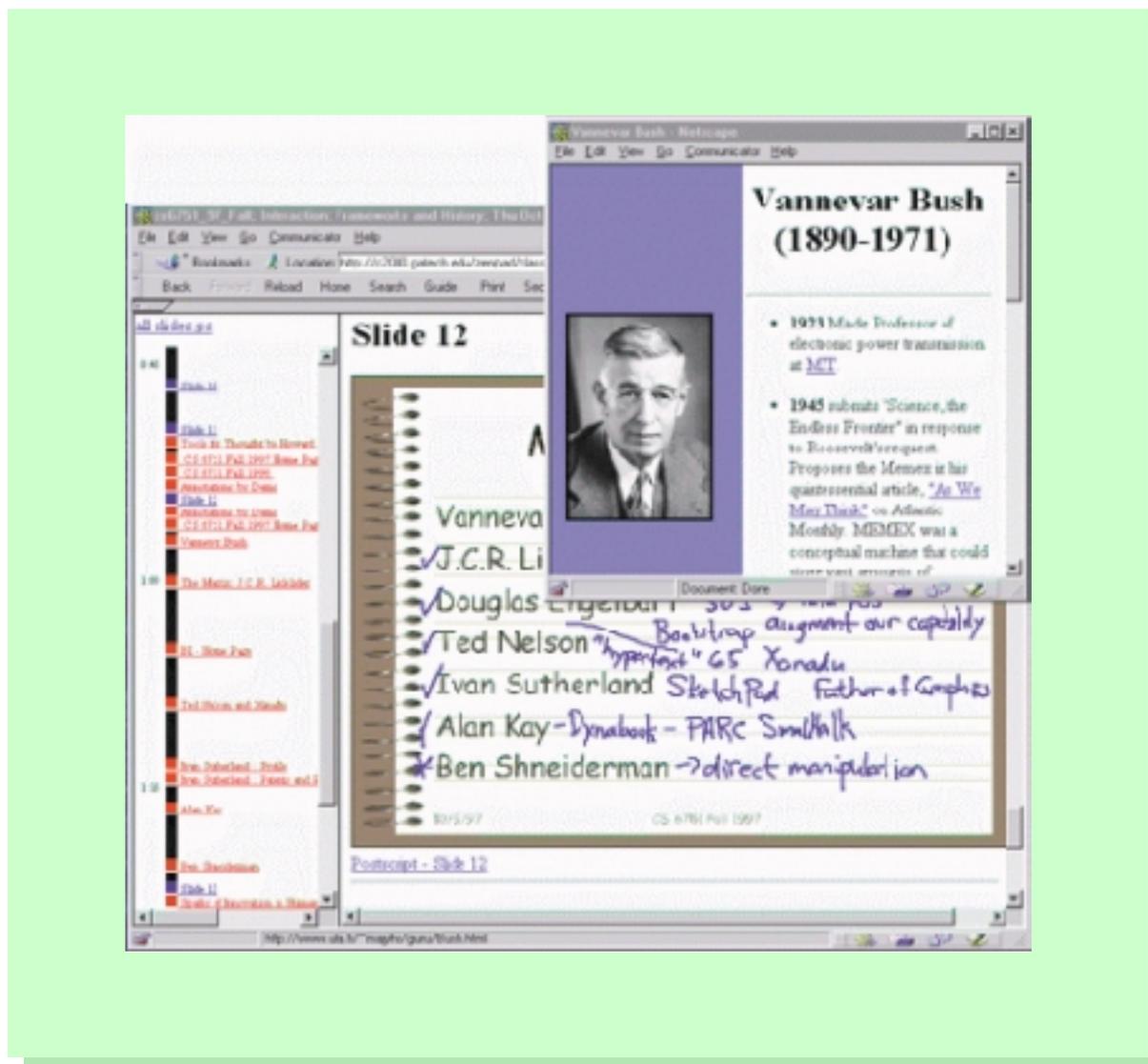


level of granularity than slide visits. We now had time-stamp information for every pen stroke (indeed, we had it for every pixel drawn), and it seemed a shame not to take advantage of that information. We wanted students to have the opportunity to click on some of the lecturer's handwriting and get to the lecture at the time it was written. We also had begun capturing Web browsing behavior during classes, so that more than just the lecturer's notes could be made available to students after class.

To provide a robust HTML-only access interface that linked individual pen strokes to audio, we created a new postproduction utility, StreamWeaver,⁹ that made use of client-side image maps supported by all of the popular browsers. The revised access interface is shown in Figure 6. This robust HTML inter-

face was an immediate success with students. We used the concept of a time line on the left-hand side of the notes to indicate the progression of activity in the lecture. This time line was decorated with the activities of the electronic whiteboard stream as well as the stream of URLs visited by the browser in class. When these two streams of activity are merged into one time line, the decorations are used to indicate when a significant event occurred for that stream, either a new slide being visited or a new URL being browsed. Blue decorations (labeled with the text "Slide *x*") indicated various slides being visited; selecting that decoration would cause the right frame to scroll to that slide in the lecture. Red decorations (with the title text from the URL) indicated a Web page being visited; selecting that decoration would load that page in an independent window.

Figure 6 The HTML-only access interface produced by the postproduction utility StreamWeaver. This interface visualizes Web and whiteboard activity captured from a lecture.



The Fall 1997 quarter marked the first time that we were able to support multiple classes with a robust capture and access interface throughout the quarter. Our main intention in this quarter was to gather good qualitative and quantitative information on how the system was used by the students and how they perceived it affecting the live lecture. The general reaction of students was very favorable, but the system was still viewed by teachers as involving too much

additional effort, so we focused our attention on streamlining preproduction and live capture activities.

Although the Zen★ system and StreamWeaver post-production utility were fairly robust, there still was much manual effort needed to coordinate activities. For example, effort was needed to take a prepared presentation and import it into the system. We

Figure 7 The Zen★ system in use with three displays. The rightmost display is the interactive ZenPad whiteboard. The middle display is a ZenViewer client set to display an overview of up to the four most recent slides. The leftmost display is used to show Web pages or more of the extended whiteboard.



needed to synchronize the beginning of the lecture with digital encoding of audio and the backup DAT recording. After class, the log of visited URLs and the digitally encoded audio needed to be placed in the central data repository, a UNIX file system. Finally, StreamWeaver had to be invoked manually to produce the enhanced HTML notes, and a link to the postproduced notes had to be made available to the students on the relevant Web page for the course.

Furthermore, we did not want to force a teacher to perform all of these manual tasks. In lecturing, all of the teacher's attention needs to be focused on delivering the lecture. The more we distract from the flow of the content in the lecture, the more we hinder the teaching process. This includes occupying the

teacher's time just before and just after the lecture, when students are likely to want to approach to ask questions. A focused engineering effort created a system that by the Spring 1998 quarter was much more automated and streamlined from the teacher's perspective.

Additional features we provided by Spring 1998 were a larger scale electronic whiteboard and video augmentation. The latter feature was a simple extension of the audio augmentation; however, the larger electronic whiteboard provided a noticeable qualitative improvement to the classroom experience.

Electronic whiteboards have been manufactured to be a maximum size of 72 inches diagonally. Whereas

this is an appropriate size for a meeting room, it is far too small to be the only whiteboard in a lecture room for 40 students. Standard whiteboards in a classroom this size are at least three times larger, and most of that space is used by the teacher. The effect we wanted to create with the Zen★ system was a collection of whiteboards such as one would see in a large lecture hall. As soon as the lecturer would fill up one whiteboard, that board could be pushed away to be viewed by the class while the lecturer went on to fill another board. With that vision in mind, we extended Zen★ to include a Java applet, called ZenViewer, that would play the role of the customizable extended whiteboard. Any number of ZenViewer clients can register with the ZenMaster server, indicating by IP (Internet Protocol) address which ZenPad whiteboard session they wish to view. When a ZenPad session was started at that IP address, the ZenMaster would then broadcast any activity from the ZenPad session to all ZenViewers. The ZenViewer client could be configured to display some part of the lecture relative to the main ZenPad session. For example, one ZenViewer could be set to display the current ZenPad slide (for remote viewing), the slide before the current ZenPad slide, or any number back from the current slide. Another mode that many teachers liked was an overview mode, as seen on the middle screen in Figure 7.

With the expanded size of the electronic whiteboard, the Zen★ system was now able to compete quite favorably with a traditional whiteboard, in terms of presentation power and (almost) ease of use.

Spring 1998 evaluation. By Spring 1998, the project had matured to the point at which we could do more serious evaluation for a larger number of courses. While it is still difficult to quantify the exact learning and teaching benefits of educational technology, it is possible to gain an understanding of how the technology is used in practice and whether its users perceive its use as beneficial and desirable.

During this quarter we surveyed students from eight different classes. Seven of the classes were taught in a Classroom 2000 environment and one was taught in a regular environment, providing a control class for comparison. We split our evaluation results into three categories—qualitative results mainly derived from surveys of the students, quantitative results derived from an analysis of the actual usage statistics maintained for the system throughout the quarter,

and a comparative study of the impact of capture on student performance.

Qualitative results. Survey results from 132 respondents across the eight different classes indicate very favorable reactions to the perceived usefulness of Classroom 2000. Table 2 summarizes the student reactions to their experience with the overall Classroom 2000 system. The answers reveal an overall favorable impression of the system and its perceived value to the students.

A number of points brought out by the questionnaire deserve comment. First, you can see that students are divided on whether or not the availability of captured lecture notes encourages skipping class. Audio augmentation was appreciated by a majority of students (63.3 percent agree or strongly agree), but video was less widely popular (31.5 percent). Despite its lower popularity, we did not choose to abandon video augmentation for two reasons. First, there was no substantial extra cost to us to produce the video, so there was no real cost vs benefit argument. Second, and more important, analysis of the usage logs indicated that students with higher bandwidth network connections chose the video augmented notes more often than the audio notes. With the promise of increasing network bandwidths, we believe video usage will be at least as popular as audio.

Another interesting set of observations concerns the note-taking practices of students who have come to rely on the automated generation of notes from Classroom 2000. When asked to describe their note-taking strategies in classes not using Classroom 2000, the following categories of note taking emerged:

- Literal note takers (35.2 percent). These students indicated that they would write down everything that the lecturer wrote down, including notes on prepared slides, as well as significant points spoken but not written down on the whiteboard.
- Copiers (29.5 percent). These students indicated that they would write down whatever the lecturer wrote down, including notes on prepared slides.
- Outliners (17.1 percent). These students indicated that they took few notes, mainly points that they considered important to the lecture.
- Listeners (3.8 percent). These students indicated that they did not take any notes.
- Other (14.3 percent). These students did not fit into any of the above categories based on their responses.

Table 2 Results of student survey from Spring 1998 (132 respondents)

Statement	Strongly Agree (%)	Agree (%)	Neutral (%)	Disagree (%)	Strongly Disagree (%)
<i>The technology made the class more interesting.</i>	34.1	37.1	18.2	6.8	3.8
<i>Capture notes helps students pay better attention during lecture.</i>	27.5	41.2	17.6	9.9	3.8
<i>I prefer a course that uses Classroom 2000.</i>	43.1	39.2	11.5	3.1	3.1
<i>Audio was valuable to me.</i>	21.9	41.4	24.2	9.4	3.1
<i>Video was valuable to me.</i>	11.8	19.7	41.7	16.5	10.2
<i>Printing slides was valuable to me.</i>	31.3	36.7	24.2	7.0	0.8
<i>Classroom 2000 encourages you to skip class.</i>	9.5	30.2	34.1	23.0	3.2
<i>Availability of notes made me less worried about missing class.</i>	12.3	50.0	17.7	15.4	4.6
<i>I expect to access notes in the future.</i>	15.5	30.2	34.1	15.5	4.7
<i>I trust captured notes will be available after every class.</i>	23.4	51.6	18.8	6.3	0.0

This breakdown of note-taking styles was roughly equivalent for students in the control class that did not have captured notes made available to them (values of 48.8 percent, 17.1 percent, 22.0 percent, 9.8 percent, and 2.4 percent for the same categories for 41 students in the control class).

When asked what their note-taking practices were in Classroom 2000-enhanced classes, 25.6 percent indicated no change in practice, 40.5 percent indicated that they no longer took notes in class, 26.4 percent indicated that they took fewer notes with more summary information, and 7.4 percent indicated some other effect, including that they took more notes. In the control class, 66.7 percent indicated no change in practice, 8.9 percent indicated that they no longer took notes in class, 20.0 percent indicated that they took fewer notes with more summary information, and 4.4 percent indicated that they took more notes. Even in the class without capture, students admitted to a change in which they took no notes. Clearly, there are other influences beyond the technology that cause students to change note-taking practices. But the trend to taking fewer notes was much more pronounced in the capture class, when compared with the control class.

From this evidence, we conclude that the system did have the overall effect of encouraging students to take fewer notes. We are not advocating that taking no notes is a good learning practice. However, plenty of research evidence supports the belief that copious note-taking practices are not good and that sum-

mary notes that attempt to synthesize information in a student's own words is a good practice.⁷ That Classroom 2000 clearly encourages students away from copious note-taking practices is viewed as a positive effect.

Quantitative results. Careful analysis of the logs for the Web server that hosted the captured notes allowed us to determine the access patterns of the students who were in Classroom 2000-supported classes. Figures 8 and 9 reveal some of the trends in usage that we observed. Each of these plots indicates when students were engaged in a "study session" using the captured notes. A study session was defined as a request to load a top-level page for any given lecture. Since we had no way of indicating when a study session ended, we assumed that if there had been no activity from a student browser for over 30 minutes, the study session had ended. This threshold value was calculated using a technique described in the MANIC (Multimedia Asynchronous Networked Individualized Courseware) project at the University of Massachusetts at Amherst.¹⁰

The top graph in Figure 8 shows a single student's access behavior for one course (an undergraduate course on software engineering taught by the author). The vertical axis represents lecture dates. This class met on Mondays, Wednesdays, and Fridays, and horizontal lines are used for clarity to indicate that day's lecture throughout the quarter. The horizontal axis represents study dates. A diamond in this graph indicates that on a particular study day (horizontal axis)

Figure 8 Comparing one student's access behavior (top) to the access behavior of an entire class

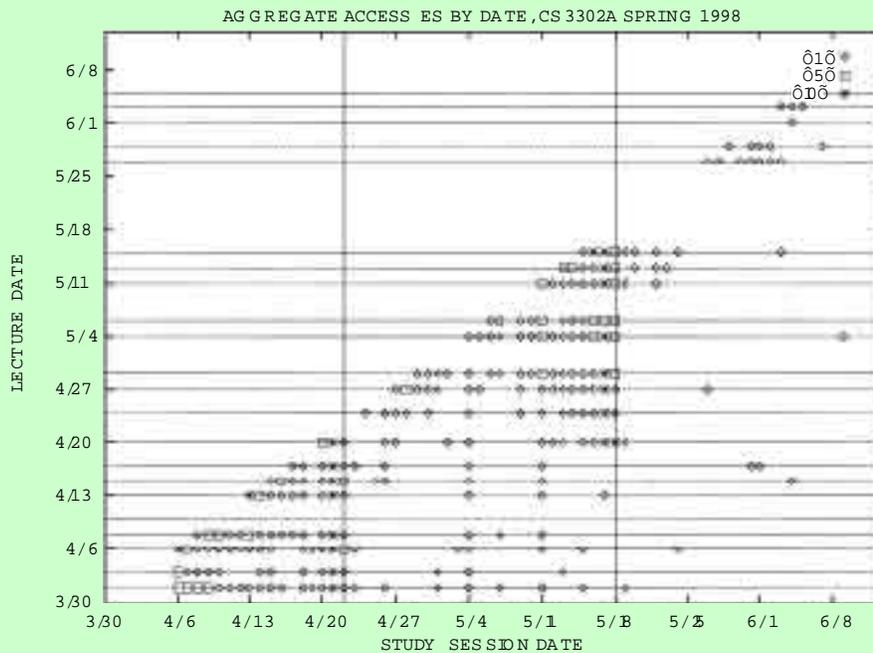
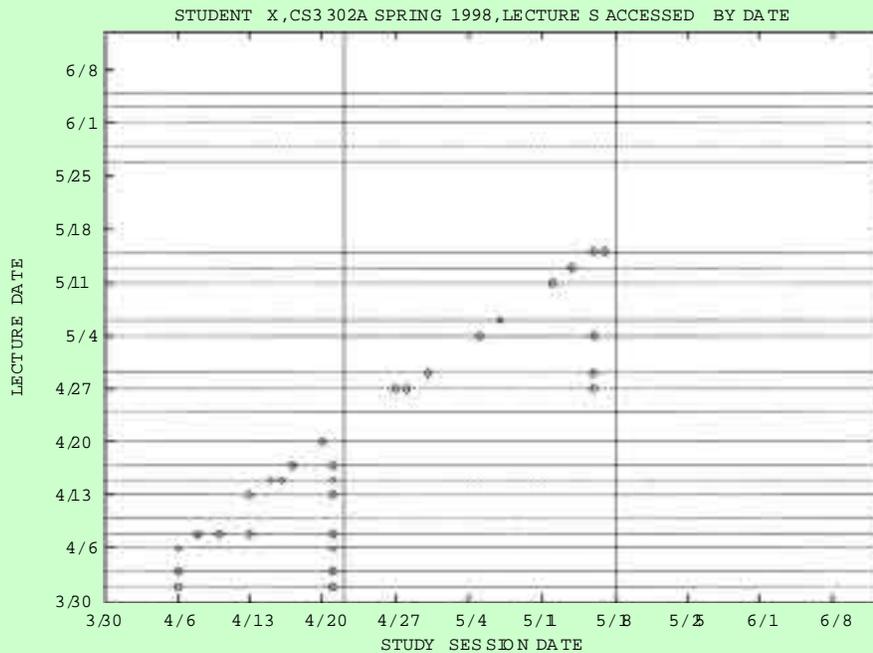
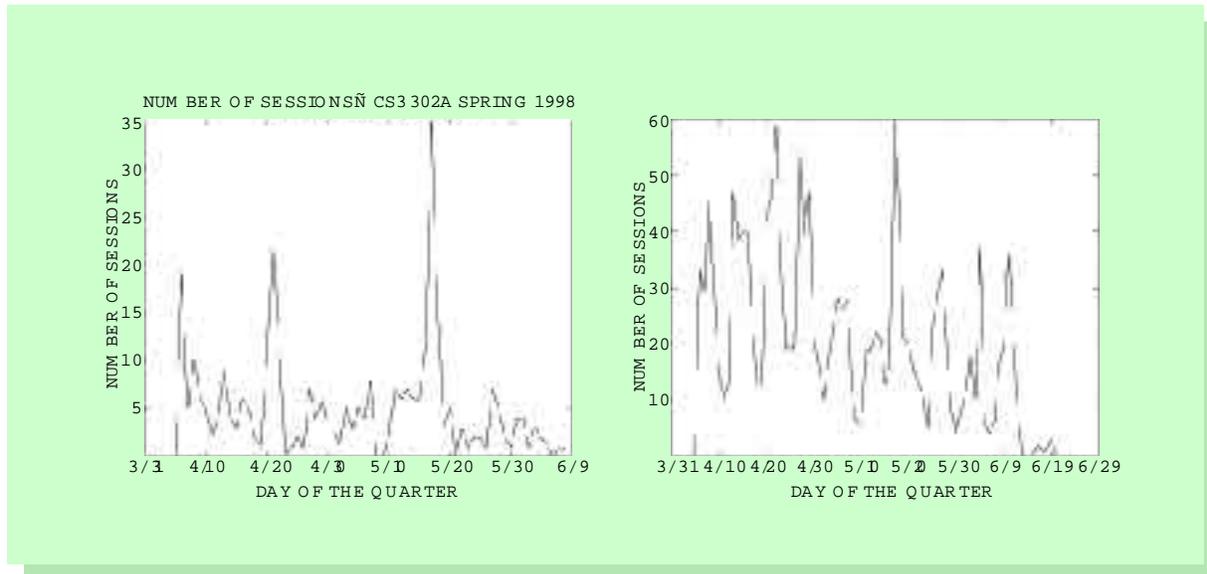


Figure 9 Comparing one course's access behavior (left) to the access behavior of all courses (right) for Spring 1998



a particular lecture (vertical axis) was visited by the student. Given this interpretation, the graph should contain access points (diamonds) only in the lower triangle of the coordinate space, as a student cannot access a lecture given on some future date.

In this particular class, there were two interim exams and a take-home final exam. Solid vertical lines indicate exam dates (4/22 and 5/18) and these lines help to interpret the graph. The first exam covered material in lectures from the beginning of class and the second exam covered material from lectures between the two exam dates. This student was in the habit of accessing the captured notes on the day of the lecture, then accessing most lectures for review a day or two before the exam.

An aggregate of this same class, which shows session behavior for all students, is shown in the bottom graph of Figure 8. Here each "cell" is further decorated to indicate the number of accesses to a given lecture. A diamond indicates between one and four sessions on that day, a square indicates between five and nine sessions, and an asterisk indicates more than ten sessions. Here we can see the "cramming" tendency of the students, because the majority of accesses to all lectures occurred on dates near the exams. We can also see that there was very little ac-

cess to lectures after the second exam, due in large part to the nature of the final exam.

What is encouraging is the characteristic usage pattern over a quarter, typified by the peaks at exam times. While it is not surprising that students would access study aids mostly before exams, it is a good sign that this behavior is repeated at more than one point during the quarter. Had students determined that accessing the captured notes was not useful, we would not have expected to see a second peak. The left graph in Figure 9 shows this trend more clearly, for one software engineering class. The right graph in Figure 9 shows the access behavior across all classes during the same Spring 1998 quarter.

A simple comparative study. During the Spring 1998 quarter, the author taught two sections of a single undergraduate software engineering course. One section was taught in the prototype Classroom 2000 room, with the usual captured lecture notes made available to students. The other section was taught in a room with similar presentation capabilities but without the ability to capture audio or video. In this second section, captured lecture notes were not made available to students. The purpose of this experiment was to determine if the presence or absence of captured lecture notes caused any noticeable differences.

A fuller description of this study will be published in a different forum, but here some of our preliminary findings are summarized:

- *Exam performance was not affected.* The average scores on two exams were virtually identical between the two sections.
- *The quantity of note taking differed drastically.* In each section, we collected all notes from 15 students and we looked over all of these notes and classified them according to the amount of note taking they represented, from none to light to heavy. As one might have guessed, the quantity of notes taken in the section that had access to captured notes was dramatically less than in the other section.
- *Students expressed a strong preference for captured notes.* More precisely, students who were denied access to the captured notes, and knew that the other section had access, were very disappointed. Our own survey as well as the teacher evaluations administered by the university reflected a very negative reaction by the students who felt deprived of this service.
- *Attendance was not affected.* Each section maintained an average of roughly 90 percent attendance throughout the quarter, though it is likely that attendance was affected by the instructor's tendency to give unannounced quizzes in class.
- *We did not track other qualitative activities effectively.* We attempted to have students fill out a journal of their study activities during the quarter, but this part of our experiment was not effective. We cannot say, therefore, whether students in the Classroom 2000 section studied more or less. All we have is anecdotal evidence that these students felt liberated from note taking and felt more free to engage in the class lecture.

These results are inconclusive and they point to the need for more evaluation of the effect of captured notes on learning and teaching, which we are continuing. We are bolstered by much anecdotal evidence that points to the perceived usefulness of the system. More and more, teachers are requesting to use the services of Classroom 2000. And many students openly tell us how captured notes have become an essential study aid and "safety blanket."

Stage 3: Recent enhancements

In this section, some advances in automated capture that have taken place in the past year are briefly discussed.

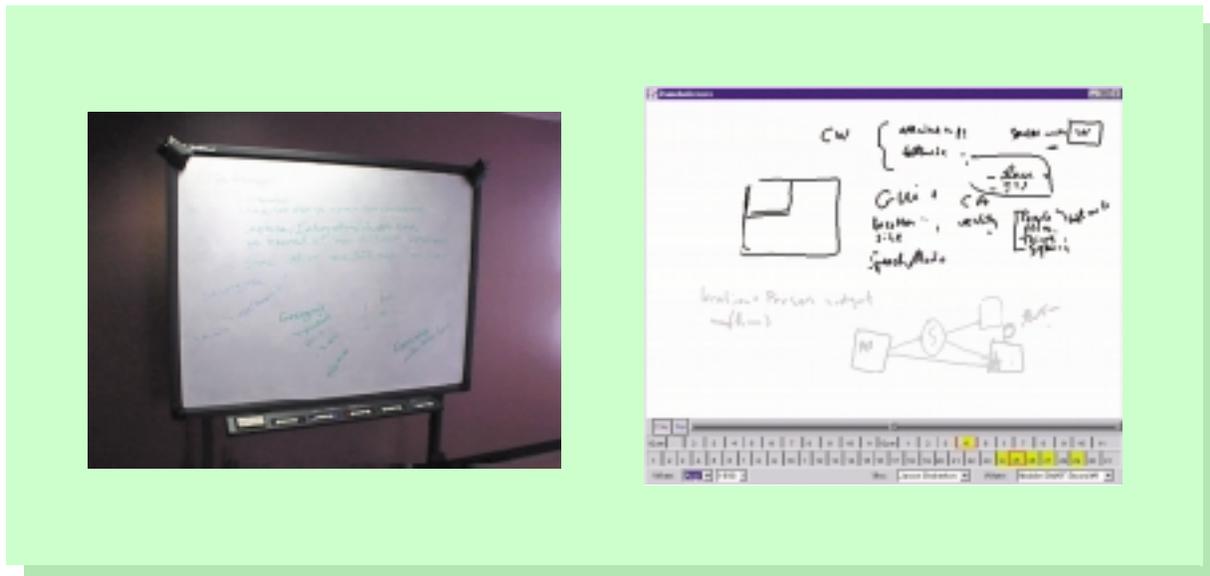
Supporting nontraditional classroom use. There have been some emerging uses of the capture capabilities of Classroom 2000 that were not well supported by the Zen★ system. These include supporting more discussion-oriented classes and supporting more informal meetings. Constraints of the traditional classroom lecture—well-defined session boundaries and one dominant speaker—were used to advantage in automating the system. When those constraints were violated, the system provided poor support.

The solution for discussion-oriented seminars was relatively simple. In the lecture room environment, we had optimized the audio setup to record what the lecturer was saying at the expense of recording what the students were saying. This was not ideal, because it was sometimes difficult to hear recorded questions, but it was a reasonable compromise for the traditional lecture. In discussion-oriented classes, all speakers are equally important, so we instrumented a smaller meeting room and gave it the capability of recording all discussion in the room with equal sound quality.

Capturing more informal meetings is less trivial. We designed a completely new system, called DUMMBO (Dynamic Ubiquitous Mobile Meeting Board), using a nonprojecting SmartBoard with an attached sound system, shown on the left in Figure 10.¹¹ When anyone approached the board and picked up a pen to write, the board automatically began to capture the writing and discussion. After a certain period of inactivity, recording would stop. All of the captured data were entered into a central database. Accessing this collection of unstructured meetings was done with a Web interface that allowed a user to browse through a time line of activity at the board, shown on the right in Figure 10. Once an appropriate time period was selected, the access interface would allow the user to replay the whiteboard activity, synchronized with the audio.

Capturing more content. Now that we have demonstrated our capability to capture a large number of classes during one quarter, we want to take on the challenge of understanding the content underlying the lectures. Students have consistently asked for this capability as a way to relate information across an entire course. The access capabilities we previously provided were optimized for playback. Simple playback is effective, but only after the students can locate the point at which to initiate playback. As more and more information is captured over a student's

Figure 10 On the left is the DUMMBO recording whiteboard for informal and opportunistic meeting capture. On the right is the Web-based access interface for DUMMBO, which allows a user to browse through an interactive time line of captured sessions to locate and play back a session of interest.



career, finding the salient points within a course or across a number of courses becomes more of a challenge. To facilitate search, we needed to have a better understanding of what was being presented in a class. We have begun experimentation using commercial speech recognition software to generate time-stamped transcripts of lectures. This technology is not yet able to produce readable transcripts. We can get 70 to 80 percent recognition rates on real lectures. Additional manual effort has produced more accurate time-stamped transcripts. We have constructed a simple interface that allows keyword search and delivers pointers to portions of lectures in which the keywords were spoken. This same technique can be used on a number of other transcripts, such as those derived from recognizing the handwriting on slides, extraction of text from prepared slides, or word search on Web pages browsed during class.

Another way to facilitate search through a large body of class notes is to provide the students with the ability to personalize the electronic notes. In other words, allow the students once again to take their own electronic notes in class. With new video tablet technology on the market, it is now possible to provide students with a note-taking environment, at their desks, that is networked and is at least as powerful as the

electronic whiteboard at the front of the class. We have been through several iterations on StuPad, a new student note-taking environment shown in Figure 11, that allows the student to view and annotate a number of different streams, both public and private, during the class.³¹ There is a separate access interface that synchronizes the playback of all captured streams together with audio and video.

Both speech transcription and student note taking were incorporated into a live environment in the Winter 1999 quarter, and we expect to be able to report on their effectiveness within the year.

Lessons learned

As well as evaluating the effect of automated capture on university teaching and learning practices, through this project we have learned valuable lessons. Here are some aspects we believe are key for successful ubiquitous computing research:²

- There should be a motivating application. In the words of Mark Weiser, applications are the whole point of ubiquitous computing.³
- The system built should address some notion of scale. Scale in this case can refer to the physical

space covered, the number of individuals involved, the number and variety of devices supported, or the amount of time over which an application is run.

- The system should be subjected to real and everyday use before it can be the subject of authentic evaluation.

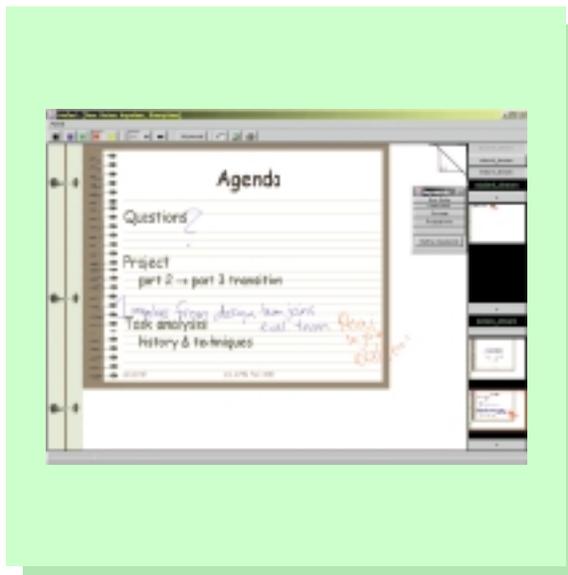
All of these criteria have been met in the Classroom 2000 project, but it has not been easy. Along the way, a number of other lessons have been learned, which are highlighted here.

It is important to prototype ubiquitous computing solutions rapidly. There is very little experience reported on successful strategies for implementing a ubiquitous computing system. It is very hard to predict what a compelling application might be, particularly since a novel application is likely to coevolve with its user population in unpredictable ways. Therefore, the researcher should deploy an application as quickly as possible in order to determine which features of the system are important and which are not. For example, by forcing ourselves to create student capture units early in the project, we learned that to be successful those units had to be able to do more than the teacher's electronic whiteboard and, therefore, had to be at least as powerful as the electronic whiteboard. Since at the time that was too expensive, we cut our losses quickly and focused on a better system to capture the lecturer's activity, which was considered useful by students and teachers alike.

The structure of an evolving system matters. Any reasonably experienced system designer knows that the early decisions made in organizing a large system greatly affect its longevity. We were very fortunate to have chosen the four-phased organization of the capture system (preproduction, live capture, postproduction, and access). One reason why this architectural scheme works is that it encourages a separation of concerns and clear interfaces between distinct phases. For instance, as long as the data interface between the preproduction and live-capture phases does not change, we are able to change one without affecting the other. Early on, we settled on a format for the data repository of captured class information, and its encapsulation has allowed us to introduce major functional changes in the live-capture phase while still maintaining minimal preproduction requirements for the teacher.

Another good example of the importance of this separation is in distinguishing between the capture and

Figure 11 Screen shot of the current student note-taking interface



access phases. The capture phase can be tailored to the controlled and unique capabilities of the instrumented environment, whereas in the access phase, the lowest common denominator of interactive and network capabilities must be kept in mind. To date, the live capture system of Zen* is far more complex and changing than the rest of the system. It is not clear that we have made the correct decisions in designing that piece of the system, but we are able to completely modify live capture as long as we adhere to conventions for interfacing with the preproduction and postproduction systems.

This lesson also points to some future research work in the area of automated capture and access. We need to build capture frameworks that provide the right programming interface, so that others can build similar capture applications for different domains, and so that those capture applications can evolve to include more streams as more capture capabilities are available. In addition, we now see the need to provide much more flexibility in the access phase, as discussed later.

Cost is not a limiting factor. When first seeing the instrumentation of our initial prototype classroom, and hearing about how we store the audio and video from all classes, the first reaction of many is that this is

a prohibitively costly experiment. The first classroom was an expensive proposition, costing over \$200 000 to initially install. Recall that part of the motivation of the name “Classroom 2000” was that we wanted to design a system that would be practical in university classrooms by the year 2000. Indeed this is the case, and we have demonstrated this by installing different versions of the system at other locations within Georgia Tech and at other universities, such as Kennesaw State University. The main technology components in the instrumented classroom—electronic whiteboards, LCD (liquid crystal display) projection, audio and video capture—are all becoming much more affordable. A recent installation at Kennesaw State University cost less than \$15 000 (not including networking and server costs). The storage requirements for a given classroom are dictated by the quality of the audio or video captured. In our experience, the storage requirements are under 15 megabytes per lecture hour, including audio and video. This is technology that can be deployed today and our experience indicates that it is a valuable investment for a university.

Capture is meaningless without access. Whereas it is important to make the live capture as complete and transparent as possible, there is no value in doing the capture if there is no reasonable and useful access to the captured record. When we initially introduced the Zen★ system in Spring 1997, we treated access as simply another mode of the ZenPad capture client. ZenPad access was adequate in our controlled classroom environment, but in the wildly heterogeneous environment that students used to access notes, it was not usable. Creating a more robust, HTML-only access interface increased the usefulness and acceptance of the system tremendously.

On a related point, it is important in the access phase to support the user’s ability to quickly search the captured record to hone in on a segment of interest. Simple capture interfaces focus on playback only. While some students used our access interface to essentially replay an entire lecture, students also wanted to browse a series of slides to find a section of the lecture and replay only that segment. Much more work needs to be done to support more effective search techniques. One idea is to use visualization strategies to allow the user to quickly see areas of interest. Another idea is to increase the semantic content of the captured information and allow queries to locate areas of interest. We have experimented with both of these strategies, but much more remains to be done.

The more experience we have with Classroom 2000, the more we understand just how critical the access tools and interfaces are. A live capture session has a clear beginning and end. The access phase has a clear beginning, but no clear end point. Over time, the most effective interface to a captured record may change. Initially, a user may want to see the whole captured experience. Later on, the user may want to see only a portion of that captured session, and may want to have it automatically associated with other captured sessions. Therefore, it is important to provide as much flexibility as possible in developing interfaces to the access phase. We have only recently understood the importance of this flexibility. For the first two and one-half years, we relied on only one, statically defined, access interface. It was only after we created a real database of captured classroom lectures that we saw both the benefit and the long-term research problems that arise with dynamically generated interfaces to captured experiences.

For the most part, we have focused on a clear separation of capture and access activities. However, it is quite clear that while accessing a previously captured experience, the user may want to modify some information, and this modification is subject to capture as well. It is still an open question how we might effectively manage capture activities that occur during access of previous experiences. For example, what is the right way to think about modifying notes from a previously attended lecture? What timing information should be retained from both the original capture activity and the newly captured activity?

Create an environment in which users can be developers. Eric Raymond hypothesizes that the success of some open-source development efforts is due to the existence of a community of users that can build upon the system.¹² While we are not yet developing the Classroom 2000 under any open-source licensing agreement, we have experienced two major enhancements that were initially developed by other users of the system. Specifically, the initial HTML-only access interface was initially prototyped by one teacher in a little more than a weekend and then incorporated by us into the system. The other example is the inclusion of the URL logging feature, another quick prototyping effort, by another teacher, that we added into the system and then enhanced. Other users have shown us different ways to use the system than we initially intended. It is our hope that making a suitable infrastructure available to a very talented population of Georgia Tech students will

result in similar improvements from the user community.

We cannot build everything in a research environment. Although a computer science community has a strong desire to build “from scratch,” a robust system as complicated as Classroom 2000 cannot be built with the typical resources of a university research group. Also, spending time developing some of the necessary functionality is not a wise allocation of research time. It is therefore necessary to build such a system from parts built by others. The difficulty arises in getting these pieces to work together. Again, this is not a novel insight.

Our contribution in this domain of system integration is to move away from standard, but heavyweight, integration technologies, such as CORBA** (Common Object Request Broker Architecture**), OLE**/COM (Object Linking and Embedding**/Common Object Model), or Java RMI (Remote Method Invocation). Much of the third-party software that we used for Classroom 2000—the commercial streaming media services, Java-enabled Web browsers, and UserLand’s Frontier** scripting environment—do not come with the necessary infrastructure for these heavyweight integration schemes. So, we built our own integration mechanisms. One of those mechanisms was a generic remote control utility that would allow us to remotely invoke a local batch file or shell script from a remote server across the network. The other mechanism used HTTP (HyperText Transfer Protocol) and XML (Extended Markup Language) to communicate commands across a network connection.

These solutions have a much greater likelihood of being easily implemented for a wide variety of platforms. In fact, HTTP servers and XML interpreters are emerging standards that consume few computational resources, making them particularly attractive for the wide variety of large and small operating systems likely in a ubiquitous computing environment. It is a challenging research topic in ubiquitous computing to develop some standards for lightweight component integration.

Understand the difference between a demonstration prototype and an evaluation prototype. A demonstration prototype only has to work occasionally and can be tailored to a very controlled and cooperative environment. An evaluation prototype must be amenable to the lowest common denominator defining the operating environment of a potentially huge user

population. Once again, the experience of producing a robust HTML-only access interface proves this point. While it is possible to do far more interesting things with access when it is under interactive programmatic control, it is much harder to build such a system for cross-platform use. One penalty we all suffer from the meteoric rise of the Web is that we must now pay heed to building cross-platform solutions to obtain widespread use.

It is valuable to simulate automation with manual effort, but only for a short period. During the project, we subjected ourselves to the pain of manually operating a number of features of the system to ease the end user’s burden. This accelerated not only the overall acceptance of the system, but also our desire to automate the manual operations. The manual effort provided insight into possible automation, and, more importantly, incentive.

Related work

Capture of live experiences is an increasingly common theme in ubiquitous computing. In addition to the Zen★, DUMMBO, and StuPad systems described in this paper, we have also developed systems to support a mobile campus tour guide with a capture component,¹³ and a system to support the capture of software architecture rationale.¹⁴

Other research teams have used this same notion of capture, integration, and access to facilitate collaborative or personal experiences. Work at Xerox Corporation’s Palo Alto Research Center (PARC) focused on capturing technical meetings to support summarization by a single scribe, who was often not well-versed in the subject of the meetings.^{15,16} More work at Xerox PARC (the Marquee system¹⁷), together with work at Hewlett-Packard Company (the Filochat system¹⁸), Apple Computer, Inc.,¹⁹ and the Massachusetts Institute of Technology’s Media Lab²⁰ demonstrates the utility of personal note taking with automatic audio enhancement for later review. Some work has focused on capturing everyday office activity and interpreting that low-level information into a summary of daily activity.²¹ Work at the University of Kent has supported the capture of archaeological observations to support field studies in Kenya.²² More recently, the Informedia project at Carnegie Mellon University, initially set up as a digital library project to investigate sophisticated search capabilities on large bodies of video and audio, has been moving toward the ability to capture real-life experiences and search through them.²³

With the exception of the Tivoli work done at Xerox PARC¹⁶ and Stifelman's Audio Notebook,²⁴ very little attention has been paid to building a capture system for live use and evaluating its use over an extended period of time. The Xerox PARC system observed the use of capture to support specialized meetings to discuss patent applications and the study lasted two years. Stifelman conducted two separate studies with different versions of her Audio Notebook and each study consisted of a small number of sessions both capturing and accessing personal notes from a variety of situations (class lectures, reporter interviews). Neither of these two projects has reported the scale of use reported here for Classroom 2000, mainly because their systems were not engineered to facilitate the automated collection and postproduction of materials as was the case for Classroom 2000.

Projects that are related to the educational application of Classroom 2000 include the MANIC system at the University of Massachusetts,¹⁰ the DEPEND (Distance Education for People with Different Needs) system at the University of Oslo,²⁵ the Chitra project at Virginia Tech,²⁶ and the Lecture Browser system in the Project Zeno effort at Cornell University.²⁷ There are also companies that facilitate postproduction of recorded meetings and lectures, such as Eloquent, Inc.²⁸

The integration of free-form ink and audio or video is a common theme for our work and the work cited above. In the "living laboratory" section the results of a simple temporal and spatial heuristic were described for associating free-form ink with audio recorded at the same time. This heuristic was used to produce the HTML-only interface for StreamWeaver. Recent work by Chiu and Wilcox discusses ways to generalize simple heuristics into a more general technique using an interactive hierarchical agglomeration algorithm.²⁹

Finally, the instrumentation of a classroom environment is related to the kind of intelligent environments that have been developed elsewhere (see for example the work by Cooperstock³⁰). Most work on intelligent environments or smart spaces has involved automation of mundane control tasks, such as control of audiovisual equipment or lighting. In our classroom, the mundane automation involved use of context to automatically predict the class being recorded, automatic synchronization of media encoding with the start of a lecture, and automated postproduction when the active session was complete.

Conclusions

This paper has described the research and experience of the Classroom 2000 project, a capture system to support teachers and learners in traditional university lecture environments. This report detailed the three-year history of the project and how various prototypes have been used and evaluated. This is a large-scale effort in ubiquitous computing that affords us a unique opportunity to investigate a number of research problems within human-computer interaction, software engineering, educational technology, distributed systems, networking, information retrieval, computational perception, and machine learning. We have only begun to tap the potential of the project, having learned a number of lessons relating to the development and operation of a living ubiquitous computing environment to support everyday activity.

The overwhelming evidence, presented in mostly qualitative form in this paper, is that automated capture is a useful feature in a traditional university lecture environment. It positively affects the lecture experience from the perspective of the students, who no longer have to feverishly copy down the details of the lectures. Our current implementation of Classroom 2000, embodied in the Zen★ system, while not perfect, provides a trusted service without much additional imposed cost on the teacher. The evidence from more recent developments in the project point toward an even more favorable benefit-to-cost ratio.

We have a number of goals for the future of the project. We have reintroduced personalized note-taking environments with the StuPad prototype and expect to evaluate its effect on overall effectiveness of the system. One unfavorable effect of the system is that it encourages too many students to take no notes at all in class, even though those same students indicate that taking notes during the live lecture helps them to encode knowledge in their own words and reinforces their own learning styles. We hope that a system to integrate student notes with the public lecture notes will encourage a better note-taking style.

Since we are now able to capture much raw lecture content, it makes sense to try to facilitate higher-level access features beyond playback. Searching a large repository of educational material is the next important step, and we believe this can be achieved by turning raw captured data into course content that con-

tains the semantics of what occurred in the lecture. Our experiments in speech transcription are a first attempt at that higher semantic goal, but we are also interested in using computational perception techniques to glean more information from audio and video signals. We have conducted initial scalability experiments to determine the computational and bandwidth requirements of the Zen★ system as it grows to support the entire curriculum of a university. We are expanding our support to cover up to four more instrumented classrooms in the next year as an empirical validation of scalability. As more and more classes use this capture capability, we hope to conduct more qualitative and ethnographic studies of the effect of automated capture in a learning environment. We also hope to extend the synchronous, co-located capture capabilities of Zen★ to the asynchronous, remote requirements of distance education.

Finally, we hope to extend the usefulness of captured educational material beyond the boundaries of the term in which a course is offered. Nearly half of the students surveyed over three different quarters of classes indicated that they expected to access captured material after a course was completed. Supporting the needs of this lifelong learning community is an important challenge.

Acknowledgments

The work described in this paper has been supported by grants from the National Science Foundation (CAREER grant IRI-9703384, Experimental Software Systems grant EIA-9806822, and CISE Infrastructure grant EIA-9818305), the Defense Advanced Research Projects, and Rome Laboratory, Air Force Materiel Command, United States Air Force (under agreement number F30602-96-2-0229). The Classroom 2000 project has received generous support from a number of industrial sponsors, including Hewlett-Packard Company, Sun Microsystems, Inc., Proxima Corporation, NEC Technologies, Inc., Hitachi America, Ltd., and the Mobility Foundation. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the National Science Foundation, Defense Advanced Research Projects Agency, Rome Laboratory, or the U.S. Government.

A project of this size and duration requires a great deal of effort. There are many, many students and

colleagues who have helped to shape this project. The author would like to call special attention to Jason Brotherton, Chris Atkeson, Khai Truong, Janak Bhalodia, Roy Rodenstein, and Nick Sawhney for their outstanding efforts in helping to build and evaluate various versions of the system. Special thanks go to Peter Freeman of the College of Computing and Jim Foley, former director of the Graphics, Visualization, and Usability Center at Georgia Tech, for their initial financial and moral support for this ambitious project.

Further information on the Classroom 2000 project can be found at <http://www.cc.gatech.edu/fce>.

**Trademark or registered trademark of Liveworks, Inc., Microsoft Corporation, Sun Microsystems, Inc., The Open Group, Object Management Group, or UserLand Software, Inc.

Cited references

1. M. Weiser, "The Computer of the 21st Century," *Scientific American* **265**, No. 3, 66–75 (September 1991).
2. G. D. Abowd, "Software Engineering Issues for Ubiquitous Computing," *Proceedings of the International Conference on Software Engineering*, Los Angeles, CA (May 16–22, 1999), pp. 75–84.
3. M. Weiser, "Some Computer Science Issues in Ubiquitous Computing," *Communications of the ACM* **36**, No. 7, 75–84 (July 1993).
4. A. J. Demers, "Research Issues in Ubiquitous Computing," *Proceedings of the Thirteenth Annual ACM Symposium on Principles of Distributed Computing*, Los Angeles, CA (August 14–17, 1994), pp. 2–8.
5. 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," *Proceedings of the ACM Conference on Multimedia*, Boston, MA (November 18–22, 1996), pp. 187–198.
6. 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," *Proceedings of ACM Conference on Human Factors in Computing Systems*, Monterey, CA (May 4–7, 1992), pp. 599–607.
7. M. L. Monty, *Issues for Supporting Notetaking and Note Using in the Computer Environment*, Ph.D. thesis, Department of Psychology, University of California, San Diego, CA (1990).
8. J. Grudin, "Groupware and Social Dynamics: Eight Challenges for Developers," *Communications of the ACM* **37**, No. 1, 92–105 (1994).
9. G. D. Abowd, J. Brotherton, and J. Bhalodia, "Automated Capture, Integration, and Visualization of Multiple Media Streams," *Proceedings of the 1998 IEEE Conference on Multimedia and Computing Systems*, Austin, TX (June 28–July 1, 1998), pp. 54–63.
10. J. Padhye and J. Kurose, *An Empirical Study of Client Interactions with a Continuous-Media Courseware Server*, Technical Report UM-CS-1997-056, University of Massachusetts (1997).

11. J. A. Brotherton, K. N. Truong, and G. D. Abowd, *Supporting Capture and Access Interfaces for Informal and Opportunistic Meetings*, Technical Report GIT-GVU-99-06, Graphics, Visualization, and Usability Center, Georgia Institute of Technology, Atlanta, GA (January 1999). Also available at <http://www.cc.gatech.edu/gvu/reports/1999/>.
12. E. S. Raymond, *The Cathedral and the Bazaar* (1998). Available from <http://earthspace.net/esr/writings/cathedral-bazaar/cathedral-bazaar.html>.
13. G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, "Cyberguide: A Mobile Context-Aware Tour Guide," *ACM Wireless Networks* 3, 421–433 (1997).
14. H. Richter, P. Schuchhard, and G. D. Abowd, "Automated Capture and Retrieval of Architectural Rationale," *The First Working IFIP Conference on Software Architecture*, San Antonio, TX (February 22–24, 1999).
15. 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," *Proceedings of the ACM Conference on Multimedia*, San Francisco, CA (November 7–9, 1995), pp. 523–533.
16. T. P. Moran, L. Palen, S. Harrison, P. Chiu, D. Kimber, S. Minneman, W. van Melle, and P. Zelweger, "'I'll Get That Off the Audio,': A Case Study of Salvaging Multimedia Meeting Records," *Proceedings of ACM Conference on Human Factors in Computing Systems*, Atlanta, GA (March 22–27, 1997), pp. 202–209.
17. K. Weber and A. Poon, "Marquee: A Tool for Real-Time Video Logging," *Proceedings of ACM Conference on Human Factors in Computing Systems* (April 1994), pp. 58–64.
18. S. Whittaker, P. Hyland, and M. Wiley, "FiloChat: Handwritten Notes Provide Access to Recorded Conversations," *Proceedings of ACM Conference on Human Factors in Computer Systems*, Boston, MA (April 24–28, 1994), pp. 271–277.
19. L. Degen, R. Mander, and G. Salomon, "Working with Audio: Integrating Personal Tape Recorders and Desktop Computers," *Proceedings of ACM Conference on Human Factors in Computing Systems*, Monterey, CA (May 4–7, 1992), pp. 413–418.
20. L. J. Stifelman, "Augmenting Real-World Objects: A Paper-Based Audio Notebook," *Proceedings of ACM Conference on Human Factors in Computing Systems*, Vancouver, Canada (April 13–18, 1996), pp. 199–200.
21. M. Lamming and M. Flynn, "Forget-Me-Not: Intimate Computing in Support of Human Memory," *Proceedings of FRIEND21: International Symposium on Next Generation Human Interfaces*, Tokyo, Japan (February 2–4, 1994), pp. 125–128.
22. P. Brown, "The Stick-e Document: A Framework for Creating Context-Aware Applications," *Proceedings of Electronic Publishing '96*, University of Kent, Canterbury (September 1996), pp. 259–272.
23. See <http://www.informedia.cs.cmu.edu/html/enter.html>.
24. L. Stifelman, *The Audio Notebook*, Ph.D. thesis, MIT Media Laboratory, Cambridge, MA (1997).
25. T. Plagemann and V. Goebel, "Experiences with the Electronic Classroom—QoS Issues in an Advanced Teaching and Research Facility," *Proceedings of the 5th IEEE Workshop on Future Trends of Distributed Computing Systems*, Tunis, Tunisia (October 29–31, 1997).
26. M. Abrams, S. Williams, G. Abdulla, S. Patel, R. Ribler, and E. A. Fox, "Multimedia Traffic Analysis Using Chitra95," *Proceedings of ACM Conference on Multimedia*, San Francisco, CA (November 7–9, 1995). Also available as TR-95-05, Department of Computer Science, Virginia Tech (April 1995).
27. See <http://www2.cs.cornell.edu/zeno>.
28. See <http://www.eloquent.com>.
29. P. Chiu and L. Wilcox, "A Dynamic Grouping Technique for Ink and Audio," *Proceedings of Symposium on User Interface Software and Technology*, San Francisco, CA (November 1998), pp. 195–202.
30. J. R. Cooperstock, K. Tanikoshi, G. Beirne, T. Narine, and W. Buxton, "Evolution of a Reactive Environment," *Proceedings of the ACM Conference on Human Factors in Computing Systems*, Denver, CO (May 7–11, 1995), pp. 170–177.
31. K. N. Truong and G. D. Abowd, "StuPad: Integrating Student Notes with Class Lectures," *Proceedings of the ACM Conference on Human Factors in Computing Systems*, Pittsburgh, PA (May 16–20, 1999), Volume 2, pp. 208–209.

Accepted for publication March 10, 1999.

Gregory D. Abowd *College of Computing, Georgia Institute of Technology, Atlanta, Georgia 30332-0280 (electronic mail: abowd@cc.gatech.edu)*. Dr. Abowd is an assistant professor in the College of Computing and the Graphics, Visualization, and Usability Center at the Georgia Institute of Technology. His research interests include software engineering and human-computer interaction, with particular focus on mobile and ubiquitous computing applications. Dr. Abowd received a B.S. degree in mathematics from the University of Notre Dame, Indiana, in 1986, and the M.Sc. (1987) and D.Phil. (1991) degrees in computation from the University of Oxford, England. His more than 60 technical publications include a textbook on human-computer interaction published by Prentice Hall. He is a member of the IEEE Computer Society and the ACM.

Reprint Order No. G321-5703.