Rooms Take Note: Room Takes Notes!

Jason A. Brotherton and Gregory D. Abowd

Graphics, Visualization, and Usability Center
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280 USA
+1 404 894 7512
{brothert, abowd}@cc.gatech.edu

Abstract

Students in university lectures try to capture the classroom experience by taking notes. In many classes, this is a tedious process and distracts the student from concentrating on the lecture. In this paper, we present a classroom that takes media-enhanced notes on behalf of the students thereby allowing them to give more attention to understanding the material rather than trying to record it. We describe the requirements of a note-taking classroom and discuss how we met them using simple techniques and small amounts of human intervention.

Introduction

Students in university lectures typically take notes in order to preserve some record of the class experience. The note-taking process in a traditional lecture involves writing down the important concepts spoken by the professor and copying the markings written on a chalk or whiteboard. It has been argued that this note-taking activity is difficult for students and that they should instead concentrate more on the actual lecture (Hadwin, Kirby, and Woodhouse 1997). Of course, access to a record of the class is also essential for good student performance. To make matters worse, as instructors are given more technology to enhance their lectures, such as displaying Web pages or real-time simulations, much of what is presented in class is either difficult to capture with pen and paper or is too much material to copy by hand. Ideally, a student should be able to concentrate on the class and afterwards walk away with a recorded copy of the lecture experience.

The typical classroom hosts hundreds of lectures a year, yet are unable to share this information with students. A classroom should be able to provide a complete record of its lectures for future reference. By having the classroom take notes, the student is allowed a greater opportunity to concentrate on the lecture. A classroom environment that is an intelligent note-taker is a specific example of the general ubiquitous computing problem of capture, integration and access (Abowd 1996). We want the classroom to record everything that takes place during a lecture, integrate all of these events into a continuous record, and provide access of the record to students for review.

In this paper we present Classroom 2000, our efforts to build a classroom that captures a rich record of a lecture experience and bundles it together for others to see. An instructor who teaches in Classroom 2000 writes on an electronic whiteboard that records and saves each pen stroke. The instructor can write on a blank board or on top of prepared slides that can be generated by most common presentation packages. The instructor also has the option of displaying Web pages on a separate projected display. The entire class is video and audio taped. These recordings can be indexed by the ink written in class, or played continuously. Immediately after class, the entire lecture is available for review as a collection of standard HTML Web pages. Additionally, students not in class during the lecture can remotely see and hear the class and see the whiteboard notes written in real time.

The rest of this paper is organized as follows. In the next section, we present some background in the general capture, integration, and access (CIA) problem with respect to classrooms and meetings. We then present some requirements of a smart classroom and the common issues they share with any intelligent environment. Next, we detail the duties of a smart classroom before, during, and after the lecture. We then discuss how we were able to rapidly build and use our version of an intelligent, note-taking classroom using simple techniques, some contextual information, and a small amount of human intervention. We conclude with some existing problems with our system and our future plans for solving them.

Background

There has been a great deal of research related to the general capture, integration, and access theme, particularly for meeting room environments and personal note taking. Our work (Abowd et al. 1998, Abowd et al. 1996, Abowd 1996, Brotherton, Bhalodia, and Abowd 1997) has been greatly influenced by previous research at Xerox PARC in ubiquitous computing (Weiser 1993) and tools to support electronic capture and access of

The Marquee note-taking prototype developed at PARC (Weber and Poon 1994), the Filochat prototype developed at Hewlett-Packard Labs (Whittaker, Hyland, and Wiley 1994), and the Dynomite personal note-taking environment from FX-PAL (Wilcox, Schilit, and Sawhney 1997) are examples of programs which support the capture, integration, and access of a meeting for an individual. In the classroom setting, we want to capture information provided by the teacher during a lecture, so electronic whiteboard capabilities such as those provided by Xerox LiveWorks LiveBoard (Elrod et al. 1992), SMARTBoard (SMART Technologies Inc. 1997), and SoftBoard (SoftBoard 1997) are essential to our research.

The Marquee note-taking prototype developed at PARC (Weber and Poon 1994), the Filochat prototype developed at Hewlett-Packard Labs (Whittaker, Hyland, and Wiley 1994), and the Dynomite personal note-taking environment from FX-PAL (Wilcox, Schilit, and Sawhney 1997) are examples of programs which support the capture, integration, and access of a meeting for an individual. In the classroom setting, we want to capture information provided by the teacher during a lecture, so electronic whiteboard capabilities such as those provided by Xerox LiveWorks LiveBoard (Elrod et al. 1992), SMARTBoard (SMART Technologies Inc. 1997), and SoftBoard (SoftBoard 1997) are essential to our research.

The integration of text with audio or video is critical to our work, and this is a popular research theme for those interested in information retrieval and content-based access to multimedia. There are a number of research and commercial systems to align textual transcripts to audio and video archives. This linking between text and audio greatly enhances search capabilities, but it is only useful when a textual transcript is available, a process that currently requires much post-production effort. When it is not practical to produce such a transcript, we rely on the implicit relationship between an audio stream and time-stamped pen strokes or keystrokes. This time-based relationship directly supports the task of a user browsing a set of notes that was taken during a class and asking “What was the lecturer talking about when this was being written?” This form of integration is exploited by the note-taking prototypes mentioned above, as well as other work at MIT's Media Lab (Hindus and Schmandt 1992) and at Apple (Degen, Mander, and Salomon 1992).

The assumption in all of the preceding work is that the user is in charge of taking the notes and there is a single device which helps in this task. Our work, in contrast, assumes that captured information is recorded by a classroom that consists of a suite of tools and devices. Other work in CIA has focused on supporting groups of people and the presentation of multiple media streams with rooms such as Bellcore’s STREAMS prototype (Cruz and Hill 1994) and Cooperstock’s Reactive Environment (Cooperstock et al. 1995), or in the support of group meetings (Nunamaker et al. 1991, Sohlenkamp and Chwelos 1994), or in the support of group meetings in virtual meeting rooms (Ginsberg et al. 1993). Like Shneiderman (Shneiderman et al. 1995), we are living in the educational environment that we created. Our work is novel in that we are attempting to provide a completely automated and ubiquitous solution to capture, integration, and access problem for a room used in an educational setting. Contrasted with other work, our goal is to create a note-taking classroom that looks and feels like a traditional classroom. Our room is designed to support instructors with a “walk-up-and-use” mentality as well as “power users” who want to use tools that go beyond the traditional classroom.

**Requirements of a Smart Classroom**

In this section we specify a set of requirements for a smart classroom. Instructors and students place many requirements on the classroom. Since we are working in a live classroom environment, we must ensure that the system is stable and always available. Any materials generated by the classroom must be available to all students taking the class. The system must also be easy to use and flexible enough to accommodate a wide variety of teaching styles. Ideally, one should forget that she or he is not in an ordinary classroom.

**Unobtrusiveness**

The computing power needed to enable a classroom to take notes must be ubiquitous. The casual observer
doesn’t need to know if the room is simply a room or a smart room. This means that the instructor and students should be able to walk in and use the room as they would any other room. Additionally, special purpose equipment for the room should be as unobtrusive as possible. For example, students might be less likely to ask questions during class if there is a microphone right next to their face or if they hear the whir of a video camera turning and zooming in on them.

**Universal Access to Materials**

Since we are operating in an educational environment, teaching real classes, we must ensure that all students have a minimal level of access to the classroom record. This means that the presentation of the class record must be platform and operating system independent. The minimal requirements for the students to view the record are a computer, a modem or a network connection, a soundcard, and a browser.

**Ease of Use**

Another challenge we faced was that of making our system transparent to the instructor. Since our electronic whiteboard looks like a computer and not a whiteboard (see figure 1), we need to put extra effort into helping the user feel like they are using a whiteboard. Unfortunately, in our first electronic whiteboard program, the instructor had to provide such information as his name, the class he was about to teach, and small details (like the topic) of the lecture. Since real whiteboards don’t have GUIs, the classroom needs to ensure that the system can automatically infer who is using it and for what class they are using it.

**Seamless Integration and Start Up**

One of our biggest challenges in Classroom 2000 is doing all the capture and integration automatically, with minimal additional user effort. For instance, in one of our first attempts at providing a smart classroom, we had a staff of three students supervise the recording of the class, the initialization of the electronic whiteboard, and the post-production merging of all the information streams together into a set of HTML pages for students access. While all of this human intervention was tedious, it provided us with an immediate and working prototype to our intelligent environment, even if we used students and TAs to simulate the intelligence of the room. A note-taking classroom needs to provide the computational services within the room the ability to begin the simultaneous recording of all media streams and the ability to weave together all of these streams of information after the class.

**Reliability**

The equipment used in traditional classrooms is reasonably reliable. Occasionally, the chalk will break in mid-stroke, or a pen might dry up during use, but when this happens, the user is able to switch pens or re-adjust the chalk and continue writing without much interruption. More disrupting is when the overhead projector has a failure, or all the chalk or pens are missing. In these cases though, most instructors can fix these problems by replacing the overhead projector bulb, or by walking into the next room and stealing some chalk. Because we want to make the computing power in the room unobtrusive, such problems might not be so readily fixed if they fail, or if their failure causes a disruption of the class, they are not so easily fixed.

Reliability, therefore, is an important requirement for a classroom if we expect professors and students to use it. A teacher expects the equipment that she is directly using for presentation to function and will not tolerate failures. More severe is when the class has a hidden failure that impacts the student’s assumption that the class will take notes. An example of this would be a video camera going bad and not providing an input signal to the recorder. Another would be the electronic whiteboard failing to record the ink written on it. One failure of this type would be enough reason for a student or teacher not to trust the classroom.

**Robustness**

It is difficult to predict and plan for failures in the classroom; they will happen. We try to reduce the harm of failure by making the systems in the room more robust. One way to do this is to save the “state” of the classroom very frequently. This way, any catastrophic failure in the system cannot affect what has previously transpired. This alone is not sufficient because the users in the class need to know if some part of the system has failed. For example, if the electronic whiteboard cannot save its ink, it should alert the instructor that any additional ink written will not be saved.

Another way to boost robustness is to make backups of recorded material. If the system fails, it can be salvaged and/or recreated from the backups. A similar approach to this is to record everything that goes on in the classroom as an independent stream of information. In this way, there is a graceful degradation of the system in the presence of failures.

**Instructor Focus**

During a class, the instructor has the ability to work with different information streams at any time. For example, the instructor may be writing on a whiteboard, discussing a Web page, running a computer simulation, or engaged in a discussion with the class. If the classroom is taking notes on behalf of the student, then it needs to know to what the student is attending at any moment. We refine that by trying to guess to what the student should be attending, and that is estimated by the focus of the teacher’s presentation. Therefore, to provide an accurate record of the lecture, the classroom needs to know to what
the instructor is attending at any given time. In our research so far, there are at least five objects of attention recognized: the whiteboard, the Web, a videotape, a computer simulation, and a discussion with students. If we are to automatically generate a faithful record of the class, our classroom needs to know the focus of attention of the instructor between these objects.

Physical Equipment
The typical classroom is ill-suited to properly capture the experience of a lecture. At a minimum, cameras, microphones, and recording equipment are needed to capture the audio and video of the class. Additionally, if this material is to be presented via the Web or over low-bandwidth connections, a computer is needed to encode audio and video into a streaming and/or compressed form.

Electronic whiteboards are also a crucial component to our classroom. An electronic whiteboard allows for the capture of what is written during class by the instructor. This is critical because even though video cameras can be arranged to capture written information, it is typically not captured well. A student cannot easily print out the ink from a lecture if it is captured on video, nor can that student modify the ink or use it to index into other streams of captured information. Electronic whiteboards come in all shapes and sizes. A common design is large screen pen-based computers such as the LiveBoard (Elrod et al. 1992), SMARTBoard (SMART Technologies Inc. 1997), and SoftBoard (SoftBoard 1997). These systems typically have a large touch-sensitive rear-projected screen as the interface to the computer. An alternative to these systems is a pen-based tablet computer attached to a projector. This functions similarly to a traditional overhead projector. Finally, there are electronic whiteboards that look, feel, write, and erase like traditional whiteboards. These boards simply have a serial out cable that can be attached to any computer that can capture what was written.

As more and more professors are using the Web as an instructional tool in class, an additional computer with a project display is needed to further support the instructor. This computer can be also be used for the display of computer simulations or programs relevant to the lecture.

While this equipment list is not extensive, it is a substantial investment. Fortunately, more and more meeting rooms are becoming equipped with cameras, ceiling microphones, and projection technology. It would be possible, for example, to build a smart classroom with a subset of these requirements. We are currently investigating other more cost-effective configurations for use in colleges outside Georgia Tech. Not meeting certain physical requirements would result in a more impoverished recording of the class, but the relationship between specific physical devices and the effect they have on the classroom’s ability to take notes is not yet clear.

Software
An electronic whiteboard needs some additional software to help make it a smart whiteboard. This software has all of the requirements of the room in general; it needs to be reliable, robust, and easy to use. Ideally, it looks and behaves just like a whiteboard while allowing the user to do tasks that are not possible with traditional boards. Additional software is needed to record other streams of information such as the URLs that were shown during class. These programs must be network enabled so that they can all be synchronized and instantiated at the proper times.

BUILDING A SMART CLASSROOM
Entities ought not to be multiplied except from necessity. In designing intelligent environments, it is hard to resist the temptation of “multiplying entities” when in reality, some simple intelligence coupled with a little preparation goes a long way. Great strides are being made in the research of speech understanding, software agents, and vision systems and their applications to intelligent environments, but in many cases, there is much that a room can infer from just a small amount of human intervention. In this section, we briefly present Classroom 2000 and its capabilities and then show how we were able to build a note-taking classroom using simple techniques and small amounts of user intervention.

Classroom 2000
The implementation of our note-taking classroom centers on a LiveBoard, a pen-based computer running Windows 95 with a large, 62” interactive display. The LiveBoard runs a Java Applet we have written called ZenPad that communicates with a central server running on a remote machine. The instructor uses the LiveBoard both as a presentation tool and as a whiteboard. While the LiveBoard is a rather large display, it is about ¼ of the width of a traditional classroom whiteboard. We extended the electronic whiteboard by using two additional computers with their displays projected next to the LiveBoard, giving the illusion that the electronic whiteboard is really the size of three LiveBoards instead of just one. These machines run another Java Applet that allows them to view current or previous slides written on the LiveBoard.

The additional computers can also be used to display Web pages to the class. Since it is difficult to take traditional notes when viewing Web pages, we capture the URLs visited and integrate them with the slides written so that students can revisit them again after class. We do this by using the Frontier environment on the Macintosh that allows us to use AppleEvents to query the status of a browser and determine when the displayed URL changes. On a detected change, an entry is written to a table with information about time of day and the URL. This information is then sent to the central server where it is
integrated with the presented slides and ink.

The classroom has three video cameras, two of which are used to capture the instructor, with the third capturing a general view of the students. The audio is captured using six ceiling mounted mini microphones. While these microphones do a good job of recording the instructor and students, we are currently experimenting with transcribing and searching the audio. To do this, we need a higher quality recording of the instructor. For this reason, we have the instructor wear a wireless lapel microphone. All of these signals are sent to a computer that encodes the video and audio using RealVideo and RealAudio. This allows us to provide audio and video to students over slower modem connections.

After class all of the captured data is sent to the central server where it is processed and woven together using a general purpose program we have written to merge together time-stamped streams of information. This program then generates standard HTML documents that the students can access. These notes (see figure 2) consist of each slide that was written in class along with a timeline that shows the order in which slides and Web pages were visited. By clicking on a stroke, a RealAudio players is spawned which plays the audio at the time the ink was written. Clicking on the timeline also indexes into the audio at the appropriate point. Students can also search the audio by typing in search keywords.

Classroom 2000 Construction

We divided the process of capturing a lecture into four activities: pre-course work, pre-lecture work, in-class work (capture), and after-class work (integration and access). Pre-course work is a one-time effort that must be spent by a human configuring the system and providing context for each class the system. Pre-class work comprises both the effort that the instructor must perform (hopefully minimal or non-existent) and the tasks that the system must do such as starting up the recordings of the class. In-class work consists of the room paying attention to the instructor and providing recording and other services that are requested. Post-class work involves the production of a publicly available HTML artifact created by the weaving together of all the recorded class streams.

Pre-course Work

In our efforts to keep things simple, we decided to reduce program complexity by manually providing some context for the classes that would be using the room. This consists of a listing of courses that use the room, the names of those courses, and the instructor for each course. Additional information about each course includes the course's start time, day schedule, and duration. This
information is stored as a text file on a central server and is consulted whenever the user starts ZenPad, our electronic whiteboard software. The maintenance of this database file is done manually at present, but the system could be easily extended so that the classroom automates this by consulting the Web for the official Georgia Tech listing of classes. Additionally, a user database must be updated so that we can provide an authentication service for the instructors.

Pre-lecture Work
Pre-lecture work consists of everything that must happen before the instructor actually starts teaching class. This includes making sure the recording equipment is in order and verifying that there are fresh backup tapes in the video and audio recorders.

- **Starting ZenPad.** Recall that we wanted users to forget that they were using a computer and instead feel like they were using a traditional whiteboard. However, the LiveBoard happens to look more like a big screen TV than a whiteboard. Ideally, the instructor would just walk up to the LiveBoard, pick up a pen, and start using it. Currently, the instructor has to turn on the screen and start up the program by clicking on its icon. We could have used the built-in scheduler of Windows 95 to automatically run ZenPad, but this approach assumes that classes always start on time, or indeed at predictable times. If a class had an additional meeting outside of normal class hours, this would not be supported. A key idea here is that a smart room need not be completely automated; some user control (as in our case) may actually be desired. Of course, it would be even better if an instructor simply walks up to the LiveBoard and has the system recognize them and the class they want to teach. This is an easy integration with a third party system, so we don’t consider our intermediate step of manually starting the program a significant concern.

- **User Identification.** Determining the class and the instructor is simplified because of the pre-course information provided to the system. The system looks up which class is scheduled to be in the room and prepares the recording equipment so that it saves the media in the appropriate class Web space. Even though we know who should be in the classroom, we still require that the instructor pass through some authentication process by specifying a username and a password. Additionally, the system requests from the user a title for the current lecture.

- **Stream Synchronization.** Prior to starting the class, the computers that encode the audio and video must be remotely started and synchronized with the instructor’s ink and slides. The class TA is generally responsible for making sure that the audio and video recording and encoding systems are in order. Unfortunately, the loading of tapes into the recording equipment is a manual process. While this step is not strictly required, we feel that it is a necessary step to have this human intervention in order to increase the robustness of the system. In our first attempts, the TA was required to manually setup the audio/video encoders by specifying the class, date, and instructor. What was most troubling about not having the stream recordings automated was that all of the data streams needed to be started at the same time. In the beginning, there were up to 4 buttons that needed to be pressed simultaneously! We consolidated all of the buttons that the TA was responsible for into one button to control all of the hardware and one for each encoding program. Obviously, as the number of recording streams increases, the number of buttons will increase. This synchronization problem was screaming to be automated.

Our latest version of ZenPad can send messages to other Classroom 2000 programs, such as one that starts the encoders. We do this with Java and simple networking to pass control and data information to our auxiliary programs. These programs then execute a third-party program via a batch file with the corresponding parameters. An example of this is the instructor starting ZenPad on the electronic whiteboard which sends a signal to another machine telling it to start encoding the audio. In this way, the instructor can start the recording of all the media streams with just one click.

Our solution for stream synchronization was just one possible approach. One problem with our solution is that since we are using the network as a message-passing medium, we introduce network latencies. While this delay is uncontrollable, our experience has not yet shown it to be a significant problem. Another solution would be if we could guarantee absolute time across different machines- then it would not matter if each media stream was started at the same time; if the absolute start time is present, then the streams could be woven together based on an offset time.

In-Class Work
Recall that two of our requirements of a note-taking room are that it needs to be robust and that it needs to predict teacher focus. We make the system robust by saving our state often, and by simultaneously recording to backup systems. As the instructor is writing on the electronic whiteboard, she is generating ink strokes. This ink is immediately sent to a central server where it is saved. A failure in the network alerts the instructor that any further writing will not be saved, indicating that the students should start taking their own notes. A failure in the encoding programs is recovered by manually encoding the information streams from the backup tapes. Notice that in both instances of failure, our backup mechanism requires additional human effort. While we could design

---

1 The instructor has the option of setting up lecture information offline. In this case, the instructor simply needs to identify herself.
automated backup systems, it is not clear that this is what we want to do (what happens when the backup system also fails)?

Since there are many information streams produced during class, it is hard to provide an interface for access to the notes that properly conveys to the viewer what was the focus of discussion at a single point in time. Determining instructor attention helps us to provide the student with this information. Our problem is simplified by only recognizing the whiteboard, a Web page, or a class discussion being the focus of attention. Our system uses a very simple model for determining instructor focus and is based on the idea that if the instructor is (or in some cases, isn’t) modifying an information stream, then he must be interested in that information stream. When the user is writing on the electronic whiteboard, then we assume they are attending the whiteboard. If the URL stream is being modified, then they must be looking at Web pages. If the instructor’s audio stream is silent and there is no activity on the other instructor streams, then they must be paying attention to a classroom discussion. While not perfect, this scheme allows us to roughly determine the focus of attention without using person-tracking devices.

After-class Work
After the class, each program sends its information stream to a central repository where our general stream integrator, StreamWeaver (Brotherton, Bhalodia, and Abowd 1997), merges and cross-indexes them into a set of HTML Web pages for student review. This is generally available minutes after the class has ended. Unfortunately, this process is not yet entirely automated. Currently, the TA for the class is responsible for FTPing these files to their proper locations and starting the StreamWeaver program. These tasks are not difficult to automate and we plan to do this soon, but at the time of this writing, it is still a manual effort.

FUTURE WORK
We are well on our way to building a fully automated room that takes notes, but there are still several aspects that need to be addressed. Our system is not completely walk-up-and-use, but we are almost there. Another way in which we can improve our room is by making it more unobtrusive. By relaxing our restrictions on user authentication, we can simply assume that the person using the whiteboard is in fact, the instructor. This raises another small problem in that we currently require the instructor upon the start of ZenPad to provide a title for the lecture. This problem can then be eliminated by assuming that the instructor writes the title of the lecture on the whiteboard as their first markings. We could then use that information and process into a title. The ultimate in unobtrusiveness could be achieved by using a SMARTBoard as our whiteboard. Then, our system would look, feel, and write like a traditional whiteboard, with ink markers, but produce a ZenPad-like output. We are in fact building such a system, adapted from ZenPad called ZenVisible and expect to put it into use in Spring quarter, 1998.

Another improvement to the system is increased instructor awareness. We can extend this to a finer level by using gesture recognition so that when the instructor is pointing to a word on the screen, we can capture this information. Additionally, we can have the instructor provide gesture markers to the system such as indicating a main topic. Using this information an outline of the lecture could be automatically generated.

Students are also potentially information stream creators. Our system could be adapted to allow students the ability to create personal notes that are integrated with the instructors notes. Much work is needed in determining how to make these student devices seamless and non-intrusive.

CONCLUSIONS
A note-taking classroom must fit several requirements that include but are not limited to unobtrusiveness, universal access, reliability, robustness, ease of use, and instructor focus. Physical requirements will vary from room to room but include an electronic whiteboard and audio/video capturing equipment. Software requirements include a mechanism for remotely synchronizing programs, a program that captures ink from an electronic whiteboard, and a program that generates a public record of the notes using the information streams created in class.

We presented our solution to building a classroom with the above requirements that requires a small amount of human intervention, but we showed how this could be extended to become fully automated. The big win for our system is that it approximates intelligence by simple techniques and small amounts of human intervention.

ACKNOWLEDGMENTS
The authors are supported in part by NSF CAREER grant #IRI-9703384. Dr. Abowd is also funded in part by DARPA through the EDCS program (project MORALE). The authors are members of the Future Computing Environments (FCE) Group at Georgia Tech and have received financial support from a number of industrial sponsors. The work in Classroom 2000 (www.cc.gatech.edu/fce/c2000) is specifically sponsored by Proxima Corporation, Xerox Liveworks, MERL, FX-PAL, Palm Computing, Apple and the Mobility Foundation. We thank these sponsors for their continued support, both financial and otherwise. Finally, the authors would like to thank the many students and faculty within the FCE Group for their strong support and energetic enthusiasm over the past 2 years.
REFERENCES


SMART Technologies Inc., SMARTBoard. SMART Technologies Inc., #600, 1177 11th Avenue SW Calgary, AB Canada T2R 1K9.

SoftBoard, SoftBoard. SoftBoard, 7216 SW Durhan Road, Portland, OR 97224.


