Title
The ActiveClass Project: Experiments in Encouraging Classroom Participation

Authors
Truong, Tan Minh
Griswold, William G
Ratto, Matthew
et al.

Publication Date
2002-04-24

Peer reviewed

The ActiveClass Project: Experiments in Encouraging Classroom Participation*

Tan Minh Truong and William G. Griswold
Department of Computer Science & Engineering
University of California, San Diego
La Jolla, CA 92093-0114
{mtruong,wgg}@cs.ucsd.edu

Matthew Ratto and Susan Leigh Star
Department of Communication
University of California, San Diego
La Jolla, CA 92093-0114
{mratto,lstar}@ucsd.edu

Abstract. Participation in classroom settings decreases with class size and diversity, thus creating passive modes of learning, due to feelings of shyness, peer pressure, and the like. Computing technology can help by creating a "safe haven" for student participation, but the successful introduction of tools into the classroom, already a dynamic and tool-rich setting, presents challenges. We describe the design of ActiveClass, an application for encouraging in-class participation. This system has been used in two sections of a second course in computer programming, providing a rich source of insight into technology-mediated learning and how technology molds student life.

1 Introduction

The broadening umbrella of universal access to higher education has brought significant diversity and sometimes greater enrollment to the undergraduate classroom. Interesting social dynamics can emerge with increased diversity and class size. Most obvious is a widely varying inclination for students to participate in the classroom. For some, asking a question is challenging authority or simply impolite. For others, the prospect of embarrassing oneself in front of fellow students is too much to bear in such an impersonal setting. Others fear that they will hold up the class with their question. Unfortunately, the professor's lecture may grow increasingly senseless to these students. Moreover, without interaction, passive learning modes emerge, which are known to be inferior to active learning. Fortunately there are always a few students who will ask questions in any setting. But these questions may not be representative of the class's needs, nor do they create full active learning for those not asking questions. Indeed, second-order social dynamics can emerge in which the quiet students resent more talkative ones for dominating the class and—from their perspective—holding up the lecture.

Universities and their teachers have made dramatic changes in response to these and other changes, with a general trend towards asynchronous, anywhere-anytime access, helping to meet the needs of a large, diverse class population. For one, hierarchies of empathetic graduate and undergraduate TA's run smaller discussions and maintain a presence in the labs and answer questions in a number of forums. Two, course content is now provided on the web, web-based discussion boards enable democratic modes of learning, and e-mail eases access to the professor and TA's. In the classroom, on the

* This work is supported in part by a gift from HP, support from the California Institute for Telecommunications and Information Technology (Cal-(IT)^2), and the ActiveWeb project, funded by NSF Research Infrastructure Grant 9802219.

other hand, the addition of stadium seating, microphones, and LCD projectors enable teaching a larger group of people, but they do not address the social dynamics created by increased class sizes and diversity.

To fill this gap in the classroom, we have developed ActiveClass, a simple client–server application for enhancing participation in the classroom setting via mobile computing devices. The basic idea behind ActiveClass is simple: using portable, wireless computing devices, students can, anonymously, ask questions, answer polls, and give the professor feedback on the class. Every student and the professor sees these lists of questions, poll results, etc. Furthermore, students can vote on questions, raising their ranking relative to other questions, thereby encouraging the professor to give those questions precedence when answering questions.

Such a system must be intrinsically different than similar tools deployed for distance learning or in computer-equipped classrooms. In distance learning, computing is responsible for virtually all data communication, not just a portion. A computer with a large screen and reasonable computing power is almost a necessity for distance learning. A computer-equipped classroom will not be constrained by the size of the computer screen or power of the device. However, such classrooms have not gained wide popularity, presumably due to the cost relative to the benefit. Mobile devices owned by the students can be employed for uses outside the classroom, thereby justifying their cost through the additional activities that they support.

This paper makes three contributions. First, it outlines a set of criteria for the success of aiding classroom participation via mobile devices, and from it a set of user interface and software design principles for this context. Second is ActiveClass itself. Third, we report early results from a detailed observations in the classroom and across the campus. We deployed ActiveClass to two sections of a computer science class (150 and 75 students each), in which all of the students had been given HP 548 Jornadas with 802.11b access.

The rest of the paper is organized as follows. In the next section we use a detailed scenario to describe ActiveClass's functionality for both students and professors. In the subsequent section, we describe ActiveClass's design. We then describe our on-going work in understanding how technology molds student life. The paper closes with a brief discussion of related work and a conclusion.

2 A Tour of ActiveClass: Setting and Use

By way of a scenario, we introduce ActiveClass and the modes of interaction we found it to support in CSE 12. Although this scenario was constructed for illustrative purposes, all the examples here are based on actual data and experiences during the experiment.

After the scenario, we will briefly discuss other details of the classroom experience not captured in the scenario. Subsequent sections will describe the concepts behind ActiveClass's design and how it supports these interactions.

Below and in the rest of the paper, we will refer to ActiveClass's users as admins and users. An admin might be the professor, one of his or her teaching assistants (TA's), or a designated student. Users are students. ActiveClass being a web application, we often refer to its features as pages.

2.1 Setting

UCSD operates on the quarter system, with 10 weeks of classes and 1 week of finals. CSE 12 is a second course in computer programming. Difficult topics such as recursion and pointers are taught in considerable detail, and there are several programming assignments. Professor G., actually a salaried lecturer, taught two sections of CSE 12 back-to-back, the first with 150 students, the second with 75. The classes meet Tuesday–Thursday. Both class sections are diverse, including both majors and non-majors, freshmen and transfer students, etc. The room is the same for both sections and has "stadium" style seating. The projector is bright enough that the lights do not have to be turned off during lecture.

Professor G. teaches from overhead transparencies. His style is to lecture for a stretch and then take questions. He uses lots of examples. Professor G. and his course are quite popular. He is highly dedicated, and he maintains an avid crew of undergraduate teaching assistants (called tutors at UCSD) who have previously taken the course. A threaded discussion board is used for students and tutors to share information; it is mostly, but not at all entirely, used for programming assignments.

ActiveClass went into use three weeks into the term, after all student drop-adds were complete and all students had a chance to attend a meeting to receive a Jornada 548 PDA, hear about the project, and do a trial run on ActiveClass.

2.2 Scenario

Monday night. Tomorrow's lecture will be about hashtables, a difficult subject for many students, and he'll need most of the lecture time for it. To conserve classroom time, he decides to add two "off topic" polls to ActiveClass. After removing old questions and adding new information links to the information page, he opens ActiveClass for tomorrow's lecture just in case anyone wants to get a head start.

Tuesday morning. Sim walks into CSE 12 a few minutes before class is to start, pulls out her wireless PDA, logs in to the ActiveClass server, and chooses the CSE 12 session. The class's Information page comes up with notes from the last lecture and links to some ancillary material (not shown).

Not interested in looking at the new information, she decides to have a look at the Polls page. There are 2, and they are open for voting. The first one asks if she's understood the previous lecture (Figure 1). Feeling pretty confident about derived classes in C++, she clicks I understood. For the second poll, the first two times are good, so she clicks each (Figure 1). Someone else entered Thursday 4-5 as an option, but that's during her Literature class. Lecture is about to start, so she goes to the Questions page. As the page refreshes over the next couple of minutes, some questions appear about the upcoming the midterm.

In the meantime Professor G. has entered the room, pulled out his slides and booted his laptop. He logs into ActiveClass and navigates to the admin's Questions page, which summarizes the questions from the class session and refreshes every 30 seconds (Figure 2). Now if he wants to take questions during class, he can quickly have a look at his laptop to see what's going on. One of his TA's has also logged in as an admin and will actively monitor the session.

Professor G. begins his lecture, explaining how hashtables are an efficient way to search. He stops occasionally to make sure everybody understands, although he gets little response.

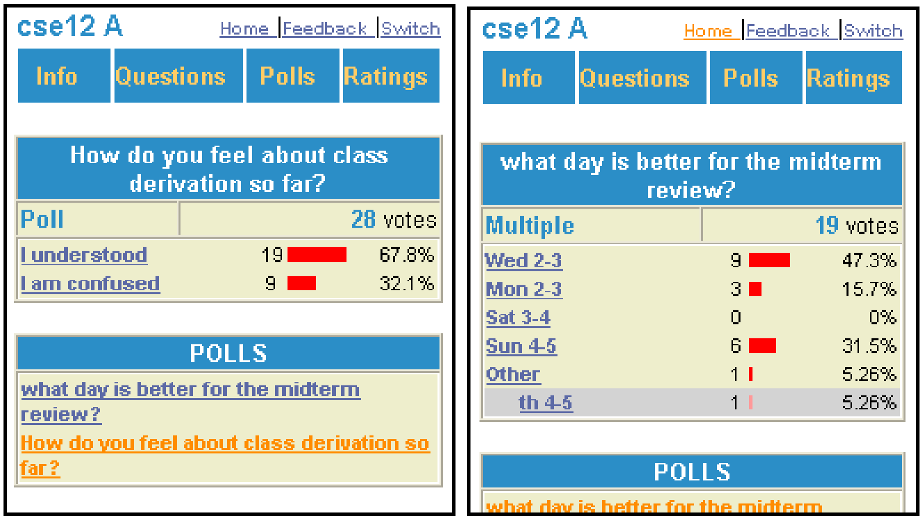

Fig. 1. A simple yes/no poll and a more complex choose-all poll.

Sim's lost. She doesn't understand why the program doesn't need to search the whole table for an element. Because nobody else seems to be lost, she doesn't want to raise her hand. Maybe he's already answered her question and she missed it while taking notes. Knowing that the midterm is coming soon, she decides she'd better ask her question through ActiveClass. Soon after asking the question she notices that many students are voting for it, and it soon rises to the top of the list (Figure 2).

Professor G. knows that at least a few students must be lost. He says, "Let's see what the virtual student has to say about hashtables," switching from his slides to ActiveClass. Looking at the top ranked question, he realizes they've missed a key concept. He draws on the students' recent homework experience with sorting to convey how keys relate to the placement of elements, and how that can help find an element quickly. The then reviews how hashing achieves the same goal without the cost of sorting. Students start raising their hands with follow-up questions. As the discussion concludes, Professor G. hides the question to reduce clutter in the view.

Sim is relieved to have had her question answered. Glad that the teacher took extra time to make things clear, she goes to the ratings page and gives the teacher a 9 and clicks Just Right for the speed of the lecture (Figure 3).

She returns to the question page, this time sorting the questions in chronological order with the newest questions at the top of the page (Figure 3). She notices that the TA monitoring the session answered a question about the due date of homework 4. She's glad he answered it here so the professor can answer more questions like hers.

With timing running out, the teacher checks his polls. He's happy to see that the students understood his last lecture; he's already behind and didn't want to speak any more on it. For the other poll, several times looked promising, but there was no overall winner. He edits the poll to only include the three best dates and asks the students to vote

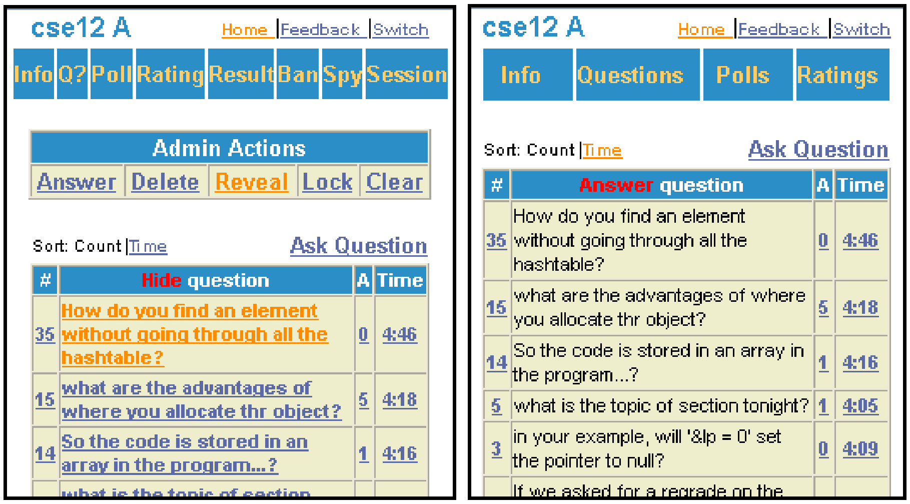

Fig. 2. The admin's and user's Question pages. The admin's page has more features, like lock, delete, hide, reveal. The list can be sorted by vote count or by time.

again before they leave the class (Figure 4). Waiting a minute, the first time becomes the clear winner and he announces that as the review time.

Now that class is over, Professor G. clicks Save to Warehouse on the Session page to capture today's questions. Thinking that one question was good, he goes to the Spy page. It lists all the questions and answers that students have entered. He clicks on the question to see who asked it (Figure 4). He goes over to his TA who has been monitoring the session, and asks him to point out Sim. As the TA nods in the direction of Sim, he says that she's always working hard in the computer lab. Professor G. makes a note to ask her to TA for him next quarter.

2.3 Discussion

Much of this scenario speaks for itself, but a few things are worth noting. The polls feature was used primarily for getting information from the class in a timely fashion without taking precious time away from the lecture. Although we had introduced the polling feature with the hopes of stimulating class discussion, stating questions in the form of multiple-choice answers was time-consuming for the professor. (There was no such problem with students coming up with questions.) Yet, the polls still proved useful in preserving lecture time. The Answer Question feature had similar unanticipated benefits. Once this feature was added (by popular demand), the TA monitoring the session would sometimes use it to answer questions that were off topic, thus helping a student while keeping the professor and the rest of the class on topic.

Before the study began, Professor G. cited many concerns about ActiveClass. For one, he was concerned that ActiveClass would be too difficult to integrate into his routine, as it would be yet another thing for him to manage during the class. However, by using a TA to monitor the session for appropriate use and the like, he was free to ignore ActiveClass until his usual breaks for questions. During these breaks, he took to calling his laptop "the virtual student". This metaphor for his ActiveClass session

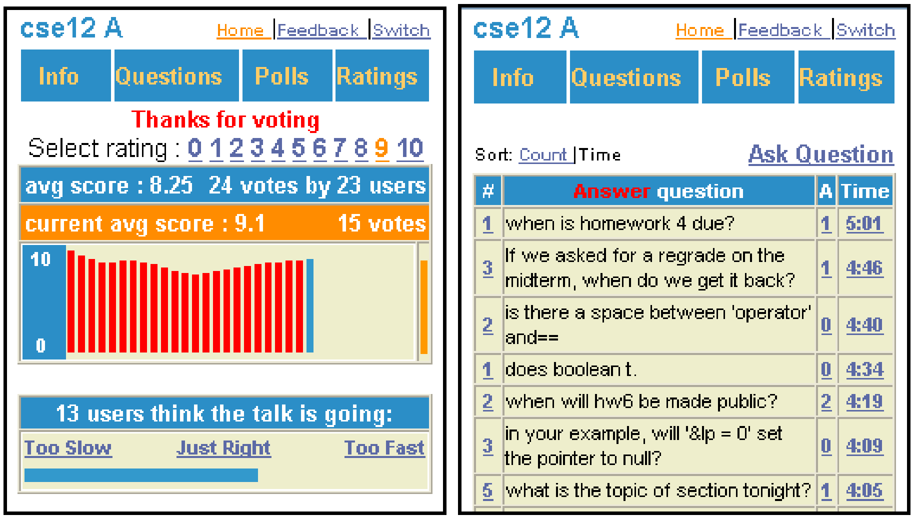

Fig. 3. The Ratings page and sorting the Questions page by time. Ratings provide feedback from the students to the teacher. There are both an average over the whole session and one based on the last 10 minutes (orange bar). The speed ratings can be seen as a bore-o-meter. Sorting questions by time highlights newly asked questions.

had two benefits. For him, it meant that his laptop was just one more student to take questions from. He would usually refer to ActiveClass only after taking direct questions from students raising their hands, clearly indicating that he preferred that students participate verbally. For the students, it meant that any apparent negative reaction to a question from ActiveClass would be absorbed by the virtual student, and no aspersions would be cast on the students. Together, the professor intended to construct as positive an atmosphere as possible for active participation.

A few other behaviors point to the possible benefits of ActiveClass. More than once the professor used ActiveClass to carry classroom activity beyond the bounds of the 80 minute lecture. He did three things. One, he carried particularly good questions from his first section to his second section of the class. Two, he carried unanswered questions from the end of one class meeting to the beginning of the next one. Three, he moved a particularly rich question offline into the discussion forum; that is, the professor used the saved state of ActiveClass as a memory aid between the class time and the time he got around to moving the question to the forum.

An interesting tendency among the students was to use both the question and answer features as affordances for unsupported communication. Before we added the Answer Question feature, students sometimes answered questions by using the Ask Question feature. Once the Answer Question feature was added, students sometimes used the Answer Question feature to thank those who provided good answers.

A few data points give a feel for ActiveClass's role in the classroom. After the novelty of ActiveClass wore off, about a third of students provided some kind of input (question, vote, etc.) to ActiveClass on a regular basis. The average number of questions asked per class session was 8, and on average 40 votes were cast per class session.

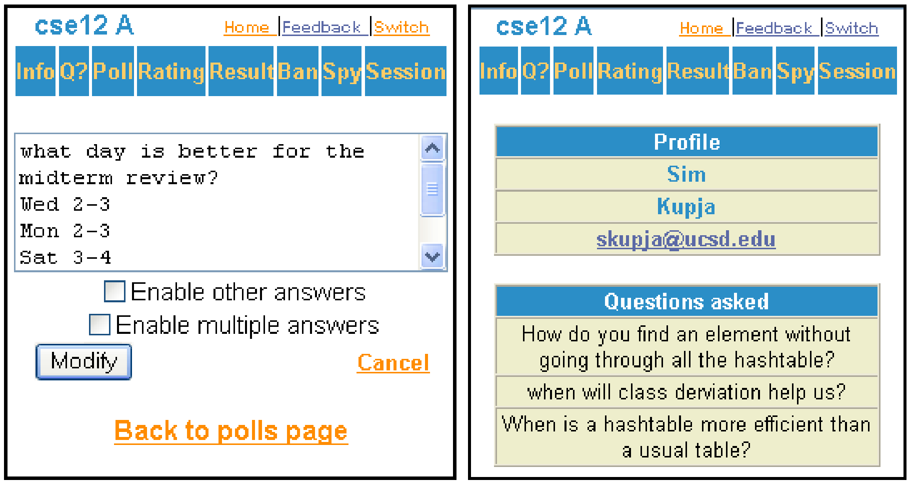

Fig. 4. Modifying a poll and "spying" on a questioner. Administrators can create, modify, lock, hide polls. The spy function enables the teacher to see who posted what.

Once the answer feature was introduced, essentially every question that wasn't directed specifically at the professor was answered by another student, with a maximum of 8 different answers for a question.

Although the level of participation seems low, it must be remembered that the professor carried over good questions to his second section, reducing the need for entering additional questions in the second class. Also, by our judgment—and that of the professor—the level of the questions was quite high and qualitatively different than seen before. After the first use of ActiveClass in CSE 12 (third week into the term), he said:

> *The most surprising aspect from today is seeing students ask questions that I don't recall ever being asked in prior versions of CSE 12. A few of these questions were especially insightful. I was very pleased to answer these questions that hadn't occurred to me, and the result is that all students were able to benefit.*

His response also points to the fact that even students who don't use ActiveClass directly are potential beneficiaries.

3 System Design

ActiveClass skates a thin line. Mobile technologies are notorious for being distracting and intrusive in social settings. A high cost to use ActiveClass technology could exclude some in the classroom. Consequently, one of our driving principles was to design it to *complement* the class lecture. In particular, we focussed on what the system could genuinely *add* to a class session, rather than replace or improve it. We also took a holistic approach to improving classroom interactions, using user-informed iterative design and limited amounts of scenario-based design to arrive at an efficacious design.

3.1 Issues

Although the ActiveClass idea is simple in principle, a classroom participation application must satisfy several conflicting criteria to succeed.

Usability. ActiveClass is targeted for a rather dynamic setting, where many things can be going on at once, not the least of which is the professor's lecture. Thus, the information on the display should be easily grasped and acted upon. Navigation should be simple and largely unnecessary. This is especially true with devices that lack keyboards, such as PDA's. Minimizing the need for navigation is complicated by the fact that small devices have small screens, reducing what can be seen at one time. In a dynamic, information-rich setting, relative simplicity can be important.

Integration into environment. ActiveClass is just one tool in a very tool-rich and custom-rich setting. Beyond being usable on its own, it must fit in with most pre-existing tools and customs that constitute the class. Otherwise, it will fail to be used, especially since it must overcome other barriers such as learning curve and "new kid" status. For example, use of ActiveClass should not preclude or complicate taking class notes. Secondarily, as a computational service, ActiveClass should consume minimal resources on user devices so that other applications are not effectively displaced.

Adaptability to emerging needs. To satisfy the norms and customs of the numerous settings that might be encountered throughout a curriculum, ActiveClass must be customizable to those settings. Generality is not a good solution, as it entails complexity (consider, for example, Microsoft Word) that is difficult to cope with in classroom settings. Also, as we undertook the design of ActiveClass, we quickly learned that it was difficult to anticipate in what way ActiveClass would be adopted by its users, and hence what features should be present to best meet their needs.

Sustainability. Educational settings are often resource poor: schools have limited funding and many needs; students, often paying for their own education, have limited resources to pay for class materials. So first, the server-side component of ActiveClass must run on modest hardware, require modest administrative skills to install and configure, and build on software that is free or otherwise readily available. Second, in order that every student have equal access in the classroom, ActiveClass must place minimal hardware and software requirements on the client device.

These criteria potentially compete with each other. For example, the requirements for usability and integration suggest an elegant and sleek client-side application, but sustainability demands ubiquity, which demands accommodating the lowest common denominator in technology. Yet, implementing a unique application to meet the needs of each device compromises both sustainability and adaptability: deploying a new feature would require updating several applications and redeploying them to hundreds of client devices.

3.2 Technology

For sustainability, we designed ActiveClass as a browser-based web application. On the client side, it uses only basic HTML to minimize incompatibilities and avoid possible performance problems (e.g., slow javascript on a PDA). The only client requirements, then, are an internet connection and a browser. On the server side we use PHP and

Apache (any web server will do), both freely available and highly portable. To avoid the complexities and potential incompatibilities of requiring a database system, all data is structured and stored in directories and flat files. The amount of data generated by ActiveClass is too small to require a database, and the content is easily represented as plain text. An added benefit is that everybody is familiar with web browsing, thus a web application with links as the control mechanism will be immediately easy for most people to use.

3.3 Architecture

Scenario-based GUI design could take us only so far in the design of ActiveClass before deploying to actual users. Thus, we invested our time to create a highly adaptable software architecture that would let us meet needs as they emerged.

We anticipated four major types of changes, each of which should be isolated to its own component. One, the representation of data (on disk) could change. Two, the behavior of the data. Three, how the data is rendered to the screen. Four, how that rendering is organized on the display (as a page). For example, adding a new feature such as Answer Question adds behavior and requires the development of a new page for display (i.e, change type four, although in fact it borrows heavily from Ask Question). Perfecting the page layout to optimize visibility, on the other hand, would involve changes to rendering and/or organization.

Our architecture is shown in Figure 5. We decided to separate changes one and two, data representation on disk and behavior, via layering because data behavior is tightly coupled to the data itself. In this design, every reference to an object retrieves the data from disk (with optional caching), builds an object for it, and get the required information from it. Due to the layering, it would be trivial to change from, say, file-based storage to a database. To separate changes two and three, data behavior and rendering, we chose to modularize them and prohibit them from calling each other. With this separation, it is possible to change the HTML rendering to, say, WAP/HDML by changing just the rendering component. To integrate data behavior and rendering, a mediator component [GHVJ95] for rendering the desired page is introduced. Taking the user's query from the web server, it is responsible for querying the required data and feeding the appropriate pieces to the renderer. This mediator component also separates changes three and four, data rendering and page organization. For example, adding an all-in-one results page for admins can be achieved by calling existing rendering functions for questions, polls, etc.

3.4 GUI Design

ActiveClass's user interface design had to take into account both the small form factor of the students' PDA's as well as the dynamic classroom setting in which they are used. Consistent display across a wide variety of devices and browsers was also a concern. Consequently, we sought a simple design that would be both easy to grasp and not touch on unpredictable features of web browsers.

ActiveClass has been designed to fit on the screen of a PDA. However, the resulting design is also appropriate for use on laptops in the classroom. The small ActiveClass window can be moved to the side of the screen, allowing for note taking, etc., to take place in other windows, with the changing content of ActiveClass still visible.

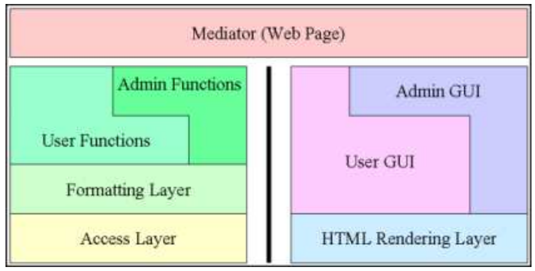

Fig. 5. ActiveClass uses mediator, layers and minimal subset architecture

In order to provide all of ActiveClass's functionality in an easy-to-use way, scrolling and navigation must be kept to a minimum. Consequently, we sought a design that would never require left-right scrolling and only require up-down scrolling if question lists and the like became too long. Also, when a scrollbar is not necessary, it is removed from the display, leaving even more room for ActiveClass content. Entering data is also tedious on PDA's and can distract the user from important activities in the classroom. Consequently, we sought modes of input that would obviate typing.

Organization. ActiveClass has several major functions, each rich in features. We use several user interface design concepts to render these functions according to our design goals. At the highest level, ActiveClass is paginated according to major categories of functions, such as questions and polls. Each function can then be seen as a rendered object, their methods being accessible through their web page.

In the PDA form factor, such a design is often insufficient to show and provide all features without excessive scrolling. In such cases, we use either abstraction or moded pagination.

Abstraction provides a small amount of information on the function's page with a link through to more information. For example, the time shown for each question is a link that opens a full description for its associated question. Typically the time of day is all that's needed, so no click is required, but clicking on a suspicious time might reveal, for example, that the question is in fact from the previous class session.

In more extreme situations, simple abstraction is insufficient because too much redundant information is involved. For example, on the admin's Question page there is no room for all the controls like hiding, deleting, etc. Moreover, as an abstraction, such links would have to reproduce much of the question page. Here then, we use what we call *moded pagination*. A set of mode links are provided that appear like a page-specific pagination bar. Clicking on one of these changes the behavior of one of the primary links on the page, but otherwise leaves the page layout unchanged. Also, to make the mode clear, a mode label is displayed in bright red. For example, when the admin is on the Question page, clicking on the Delete mode link will take the page into delete mode. Clicking on a question will delete it and the header on the question list says Delete Question(Figure 2).

Thus far, we have taken great pains to isolate moded pagination to the more complex admin pages, as moded use can result in errors if the current mode is overlooked or forgotten.

Managing User Interaction. To minimize the tedium and distraction of user input, we attempted an input model based on single-click interactions wherever possible. One design rule is that any content provided by any user should be displayed as a clickable/votable input for other users. For example, each submitted question can be voted on by a single click. This prevents users from entering the same question multiple times if an existing question is close to theirs. This not only eliminates unnecessary, tedious typing but also keeps the question list as short as possible. This same principle is applied in answers to questions and polls (e.g., choices entered under the "other" category of polls are clickable), thus, only one person ever has to invest the effort to generate an input by typing.

The one-click model has several complementary benefits, which are best demonstrated through a counterexample. In our original design for the Questions page, each question was displayed with a check-box next to it so that the user could quickly go down the page, selecting all the questions of interest and then click the "Vote" button at the end of the page. This design requires just one more click than our current design (the "Vote" click) and requires just one interaction with the ActiveClass server. However, the approach creates several design problems. First, check-boxes and radio buttons are not displayed consistently across browsers, making predictable formatting difficult. On PocketPC's, in fact, they tend to be quite large, wasting space that could be used for showing user content. Second, the "Vote" button would often be off the bottom of the screen, so even if a user was voting for just a "top" question, scrolling was required (and the button was invisible to a novice user trying to make sense of the interface). Finally, the Questions page could not be automatically refreshed, because sometimes the user would be in the middle of voting and lose his or her selections. (Frequent automatic refreshing, again, minimizes user input because the sooner that others' input is seen, the sooner that it can be clicked for further input.) Thus, a separate Results page (or results mode) was required, entailing additional interaction and possible confusion because of the moded interaction.

In summary, a page using only links on text items for interaction permits simple one-click interactions, a consistent and compact display, and automatic updating of the display to frequently provide other users' input for selection. We were able to display 20% more lines of content on the display by moving to the link-based interface. As a consequence of this design, of course, all non-clickable input (e.g., a new question) must be provided on a non-updating page, reached by clicking a link from an updating page (e.g., "Ask Question"). The resulting input pages for new questions, answers to questions, etc. are quite simple and fit on a single page.

The only problem that can be seen with this approach is that it requires an HTTP connection for every action: Voting for N questions will imply sending N request to the server. However, because we use only basic HTML, the size of the pages to be loaded is small (2KB), so the load on the server and the wait time for the user are both small.

4 Complex Literacies

To understand how mobile computing and ActiveClass impact student learning, we must understand how students interweave the usage of the PDAs into their everyday lives, including into everyday classroom practice. Thus, we are also doing an ethnography of the classrooms and, just underway, the use of the devices across campus, including in dorms, eating places, the library, the on-campus shuttle, etc. Our ethnography is an anthropological-style fieldwork approach—observing students, interviewing them, hanging out with them, and e-mailing them in order to understand the role of PDAs in their larger lives. In this, we see an important response to the problematics of ubiquitous computing. It is almost impossible to measure "before" and "after" by traditional means. Because ubicomp touches comprehensively on all aspects of life, it is by nature embedded in ongoing social processes. Below, we describe an analytic method that is designed iteratively to identify social processes (and their variations) across various settings. We use fieldwork data gathered via observation and interview that initially has the form of unstructured or semi-structured text (See Figure 6; AC is ActiveClass).

```
Instructor G. asks for questions at 4:30 - Gets one - SMCF.

SGCC#1 now using keyboard to write - glances up at G. and
then writes for a few sentences.

2nd question also SMCF.

G. asks TA for questions from AC. G. has decided to have
a TA mediate the AC questions - to look at PDA and let him
know if questions come in. (4:30)

A question came in asking if their were classes in C as well
as c++ (which cse12 uses) at 4:17.

At 4:27 an answer appeared on the questions list. So stu-
dents using questions list to communicate.
```

Fig. 6. Section of in-class fieldnote 1/24/02.

4.1 Methods

The analytic method that we use is called grounded theory [GS67,Str86]. It is supported by a software package, Atlas/ti, developed in Germany, that helps track theoretical and substantive codes, processes, and various types of data (notes, memos, photos, films, secondary documents) through its development. In particular, Atlas/ti is helping us correlate the field note data as presented above with other data sources such as ActiveClass log files.

Grounded theory is an excellent method for eliciting respondents' own categories ("in vivo codes") and for discovering the dimensions of interactive processes [GS67,Str86]. It does this by constantly comparing across situations, identifying processes in one site, and testing their dimensionality across the sites under study. The initial idea is to elaborate the theory and develop a dense model of interactions and processes; later work winnows out the central processes and identifies a usable shape for the model. It is especially useful where, as in the case of this study, the question of exactly what we are

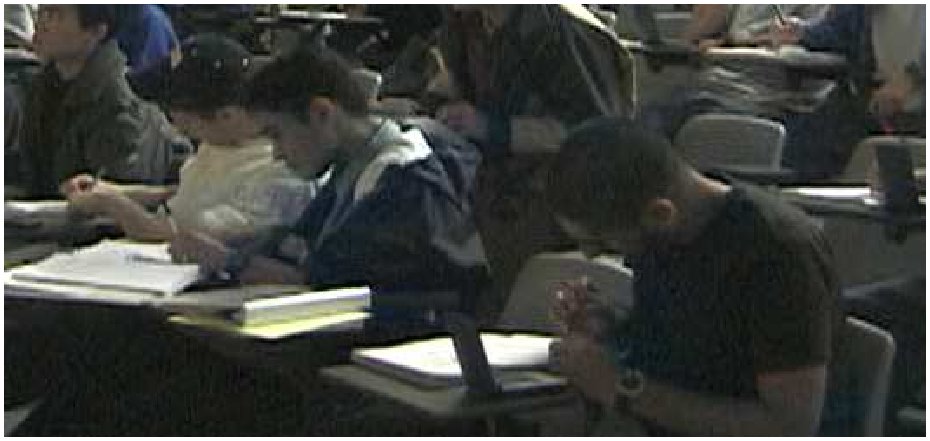

Fig. 7. ActiveClass in use. Note the multiple tools being used by students.

looking for is the first order of business. We are clearly interested in improved productivity, richer educational experiences, and ease of use with integration with extant technologies. However, the how of these complex processes remains to be discovered. In the interest of providing a useful model for others studying these processes, the facts of our success or failure are only as good as they are generalizable. In this case, we are choosing generalizability via modeling and validation across local sites. In ubiquitous computing, there is little chance to think of "before and after" or "input-output" terms. Our respondents already have laptops, some have PDAs, and nearly all have cellular phones with varying capacities. Thus we are faced with the challenge of describing the ecology of devices used, and how the Jornadas fit in with that ecology [Sta95]. In this ecology, we are not neglecting lower tech devices (notebooks, pens, television, books) in our description. Many students continued to take notes using pens, pencils, and notebooks even if they incorporated the Jornada, add-on keyboards, or a laptop into their in class activity. By including the other tools used by students we thus avoid the tabula rasa fallacy of social impact and innovation research, and garner what Geertz calls "thick description" [Gee73]. The challenge then becomes converting these descriptions into usable models. For all these reasons, ethnography and grounded theory are our tools of choice.

4.2 Preliminary Empirical/Theoretical Findings

We have currently, over a ten-week period of ethnographic data collection and preliminary analysis, garnered enough data to make some initial analytic observations. We are guided by some ongoing theoretical commitments, as well. These include:

- An interest in the materiality of the devices in question (for example, how and where they are carried, when they appear, how students handle them, where might they fit on a crowded desk or in the hand when carrying a load of books). We are especially interested in the material-cognitive juncture.
- Interest in complex literacies, that is, how web literacy meets book literacy, and how these meet other sorts of literacy, such as how to read schedules and formal

documents, how to look at buildings and maps, and the built environment more generally.

- Workarounds in the classroom and outside of it. Gasser [Gas86] defines a workaround as the work that is done when confronted with an obstacle to routine computing work. He defines three types: fitting, augmenting, and working around. How do respondents deal with faulty PDAs, leaving the wired zone without knowing it, using the PDA for unofficial reasons, or fitting the PDA with other forms of usage (e.g. instant messaging to a loved one)?
- Emotions and technology. One of the goals of ActiveClass is to encourage participation from all students, regardless of, for example, gender or ethnicity. Will shy (emotionally or culturally) students feel free to ask questions using the Jornadas in ways they do not when they have to raise their hands? What is the texture of student-instructor question and answer periods, including what is handled in class and what is deferred?

In answering these questions, we have identified several initial sorts of observations useful for the ubiquitous computing community.

Physical management of PDA in classroom setting. We use observations of students addressing the physicality of the PDA. These are ways to address practices of incorporation and negotiation that may appear less directly material, but which are by no means less durable. When students manipulate the PDA/stylus with hand/eye and desk, notebook, and pen, they in fact reveal much about the material infrastructure of the classroom. In one such example, a student managed to hold both a pen and the PDA stylus with one hand by using a kind of 'chopstick' grip. This grip allowed the student to rapidly shift back and forth between a paper notebook and the Jornada. This physical shifting is a observable marker for the conceptual work that incorporating new technology into the class requires. We have several fascinating images from video data that show students absorbing the PDA into their physical practice.

Students do not just passively use the technology handed to them, rather, they wrest the technology from its embedded presentation, and use it in the classroom context. For example, we observed students using the Jornadas to 'physicalize' web pages, notes, and other electronic data. Rather than use the wireless connection to 'beam' data to another PDA, students would bring up an URL or other information on their PDA and then hand it to another student to view. We call this process of developing unique practices 'wresting' in order to call attention to the ongoing 'sticky' nature of the students' work to transform the ActiveClass application and Jornada into tools that will work for them. Rather than a passive incorporation of the Jornada and ActiveClass, some students actively 'wrestle' with the limitations and affordances of the devices. Three insights are particularly clear from this example. First, students are tailoring the application to their own pace and sense of what is needed in the classroom. Second, students freely leave the assigned usage of the device in the interest of utilizing other resources in the classroom, something not widely seen before the advent of such devices (although certainly not absent altogether, as when a student brings a related book to class). Third, there is a seamless transformation of the hand-held device into a book-like object - while students can beam web pages to each other in the class, these students choose to show the websites to classmates and verbally explain them. This is part of creating idiosyncratic

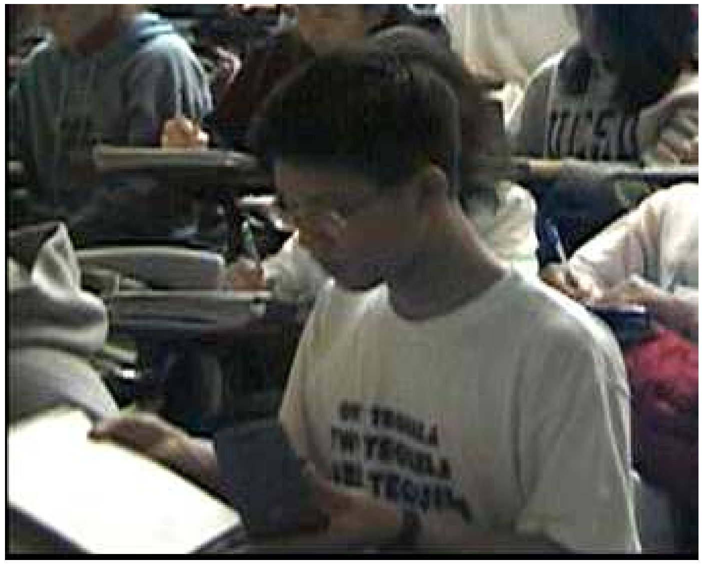

Fig. 8. Student managing notebook and Jornada.

infrastructures and ecologies of literacy. Developing applications and technologies that support the 'wrestings' of students requires that we recognize and record student innovations as well as construct infrastructures that can support novel and emergent practices.

PDA use in the classroom not oriented to ActiveClass (project or application.) Initial observation, as noted above, seems to make this a feature rather than a bug. The PDA was not particularly distracting, even when used for extra-class activity. Students already bring laptops with wireless internet into the room. Still, logs and observation reveal that although more than 50% of the students had either a PDA or a laptop out and on, numbers logged into ActiveClass are often less than this. Some students are using the devices for other activities not related to ActiveClass, but may be class-related. For example, we witnessed students using laptops and Jornadas in class to run project source code in order to test concepts and ideas from the lectures and to access course web pages. In one interaction, a student accessed a web page, and then turned to the student next to him to show him the web page.

Incorporating low participation students into class conversation. ActiveClass is a new channel of communication for 'shy' students. We have no baseline with which to evaluate whether or not previously non-participating students are using the ActiveClass forum to send in-class questions. We do have a wealth of anecdotal evidence based on our shared years of teaching, however. We expected that the anonymity of the questioning feature on the PDAs would facilitate easier questioning in this large classroom. However, based on a preliminary analysis of our logs and observations, this does not seem to be the case. Many students still raise their hands to ask questions - too many to handle in the classroom per se. However, there are two avenues by which shy students may extend their question asking outside the immediate face-to-face classroom. Students may participate in out-of-class fora such as a web board. The questions asked, on which all students vote for significance, are often deferred to the discussion sections. They thus have a longer durability than the ephemeral nature of face-to-face question asking in the classroom. We have a video clip of three students, including two Asian women, *not* using or even looking at the ActiveClass application. All three students are intently focused on the instructor, G., taking what appears to be detailed notes. Perhaps ActiveClass does not provide enough of a perceived benefit to these students, at least enough to overcome the cognitive load imposed by the technology. In part, this may be because of the focus and structure of the class is very oriented towards data gathering and remembering. The class does not orient students towards the type of reflection and consideration that are emphasized in other class types (such as more seminar style classes that are about more general topics, or some discussion sections) Students are kept very busy recording the directions and technical tips of the instructor, and are only asked intermittently to engage in more reflective cognitive tasks. In future work, we will explore this question of the durability of queries and the social-psychological nexus from which they arise. These questions will more adequately be addressed by focus group and interview data as well as by looking at the types of questions that are asked on ActiveClass, on web board, and by students verbally, and by observing discussion sections.

Redefining participation. The areas defined above seem to indicate that the incorporation of new technologies into (and outside) the classroom also mean reevaluating what we mean by participation. Other studies have shown the addition of new technologies into existing work contexts does more than just increase or decrease the adoption and use of existing practices. Rather, existing practices change and new ones emerge, in an ecological fashion. The ActiveClass project provides an excellent case for redefining what we consider to be 'good' participation. One example of how ActiveClass changes class participation in unforeseen ways is through the possibility of question deferral, as noted above. As any teacher knows, sometimes questions are asked by students that, although excellent, are too complex or off-topic to be adequately addressed in the classroom moment. Often these questions can disappear given the time constraints of the classroom (and the memory constraints of the teacher!) A new feature being developed for ActiveClass is a way of 'popping' such questions from the ActiveClass forum to a more general web board forum. Here, the questions can be addressed by the teacher, teaching assistants, or even students at their leisure. Another example of this was the emergent practice Instructor G. developed as he became more comfortable with the application. As the class progressed, G. began returning to the previous session's question queue at the beginning of each class. Questions that had not been previously answered were often addressed, providing more information for the students, as well as a greater sense of course continuity. Thus, ActiveClass can provide a way for such questions to be remembered and repurposed - deferred rather than disappeared.

4.3 Summary

As we are at the beginning stages of this project, many questions must be themselves deferred until more data are collected. At the same time, we find fertile ground for answering the questions we have addressed to date, along with some surprises. It is important again to note that this is not an "impact" study per se. Rather, along with other similar research in ubiquitous computing, we are faced with a challenge of a different order. We must define a complex situation where multiple forms of literacy interact, and where multiple devices are already in very active use [BNS+00]. Thus, rather than a computerization question, we are faced with questions about the work that students and instructors do, the nature of their communicative processes, the material environment and how that affects the emerging technology ecology, including the PDAs. Thus, our future investigations bring us outside the classroom, into the dormitories, eating places, libraries, and study halls of our students, as well as into the worlds of teachers and teaching assistants.

In this, we follow the work of Elfreda Chatman [Cha92,Cha96,CP95] who defines "information world" in a like ecological fashion. Information is imbricated in the living community, and any empirical attempt to understand its usage must take the norms and values of that community into account, as well as the work and play processes. We anticipate that the most general interest (outside the walls of academe) of our findings will be in the areas of material juggling of multiple devices, and the ensuing changes in practice and usage that result.

5 Conclusion

With the increasing class sizes and diversity seen in higher education, pressures against classroom participation, and hence active learning, also increase. Mobile computing has the potential to bring new modes of participation into the classroom.

In this paper we have introduced ActiveClass, an application designed to realize this potential. ActiveClass was designed with attention to usability, integration into the classroom context, adaptability to evolving needs, and institutional sustainability. Our link-oriented, click-driven interface, employing pagination of major functions and abstraction of more complex features permits extremely simple, compact renderings that display consistently on virtually any browser.

We deployed ActiveClass to two large sections of a computer science class, 225 students in all, in which all of the students had been given HP 548 Jornadas with 802.11b access. We followed the class through direct observation and other methods, evolving ActiveClass to meet the class's needs, as learned through emergent behavior, requests, and our observations.

From this study, we learned several things. First, our mediated and layered architecture was essential rapidly adapting ActiveClass to the class's discovered needs. There were numerous "surprises" throughout the term that reinforced our choice to design in adaptability rather than perfect requirements in advance. Second, in addition to ActiveClass being used in anticipated ways, many unanticipated modalities of use of ActiveCampus emerged during the term, pointing to genuine latent needs for some sort of assistance in the classroom. However, along with the unanticipated use of polls for streamlining class management came the failure of its use in posing genuine class questions to the students. In retrospect, it is not surprising, even reassuring, that the structure of polls seem to be too constraining for productively provocative questioning of students.

Our goal of sustainability is as yet untested. This study was only possible through the generosity of Hewlett Packard. Although device prices for PDA's and laptops are dropping rapidly, the magic price point of an expensive textbook (or two) still appears to be a ways off. From the institutional perspective, the outlook is promising. 802.11b infrastructure is increasingly affordable, and the lightweight design of ActiveClass makes it easy to administer. It is also reassuring that professor G. is continuing to use ActiveClass this term (without full coverage of his students with PDA's), and is already arranging to use it in the Fall term as well.

With just one quarter of usage behind us, we already observe that PDA's enable unique uses in the classroom, both inside and outside ActiveClass. The inventiveness of both the professor and the students with the technology, despite the challenges of handling yet another tool in the classroom, is indicative of real opportunities.

References

[BNS+00] Ann Bishop, Laura Neumann, Susan Leigh Star, Cecelia Merkel, Emily Igancio, and Robert Sandusky. Digital libraries: Situating use in changing information infrastructure. *Journal of the American Society for Information Science*, 51(4):394–413, 2000.

[Cha92] Elfreda Chatman. *The Information World of Retired Women*. Greenwood Press, Westport, CT, 1992.

[Cha96] Elfreda Chatman. The impoverished life-world of outsiders. *Journal of the American Society for Information Science*, 1996.

[CP95] Elfreda Chatman and Victoria EM Pendleton. Knowledge gap, information-seeking and the poor. *Reference Librarian*, 50(49):135–145, 1995.

[Gas86] Les Gasser. The routine uses of computing work. *ACM Transactions on Office Information Systems 4*, pages 270–290, 1986.

[Gee73] Clifford Geertz. Thick description: Toward an interpretative theory of culture. In *The Interpretation of Cultures*. Basic Books, New York, 1973.

[GHVJ95] E. Gamma, R. Helm, J. Vlissides, and R. E. Johnson. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, MA, 1995.

[GS67] Barney Glaser and Anselm Strauss. *The Discovery of Grounded Theory*. Aldine, Chicago, 1967.

[Sta95] Susan Leigh Star. *Ecologies of Knowledge: Work and Politics in Science and Technology*. SUNY Press, Albany, 1995.

[Str86] Anselm Strauss. *Qualitative Research for Social Scientists*. Cambridge University Press, Cambridge, UK, 1986.