INTERACTION DURING LECTURES USING MOBILE PHONES

Henning Bär, Max Mühlhäuser, Erik Tews, Guido Rößling
Department of Computer Science
Darmstadt University of Technology, Darmstadt, Germany
{*hcbaer,max,e_tews,roessling}@rbg.informatik.tu-darmstadt.de*

ABSTRACT

Interaction is an important part in lectures. Both educators and students can benefit from software support in large lecture halls particularly. As a cost-effective and widely usable solution, we have investigated interaction based on the mobile phones owned by most students. After presenting design criteria, we discuss a concrete prototype for mobile phone based interaction. The solution is part of an open platform that also works with PDAs and Laptops.

1. INTRODUCTION

Interaction and feedback as essential components of lectures can improve the success of learning (Waite et al. 2003). However, achieving meaningful interaction is difficult, especially in lectures with an attendance of (far) more than 100 students, such as European undergraduate computer science lectures.

The most common interaction form, hand raising, is at best difficult in large lectures. Secondly, the size of the audience effectively prevents the educator from answering the potentially large set of questions. Moreover, students sitting towards the background are often overlooked. Interaction during office hours, in forums, during lecture breaks and after each lecture are no sufficient substitutes since the perceived "pressure" to ask drops considerably, other obligations may hold students back, students and learners have difficulty to refer back to the subject, and the dissemination to the residual audience is more difficult..

Interaction covers communication among students and between students and the educator. Interaction among students during lectures has to be considered carefully, as it can easily distract from the actual lecture contents. The educator should therefore consider moderating this type of interaction.

Interaction between educators and students can be initiated by the educator, for example by posing questions, taking decision polls, or starting a multiple choice quiz. Students typically provide feedback by asking questions, commenting on the lecture style or giving general comments.

Large-scale lectures often suffer from a lack of time to address all student questions. Therefore, both educators and students can benefit from a solution that allows the educator to pick the most relevant questions to answer. For this end, the educator must be able to quickly scan the current set of questions, in order to determine the perceived relevance of a given question for the audience.

Lectures are often evaluated at the end of the term and possibly also at mid-term. This means that the educator does not have much chance to address general comments, such as "speaks too quickly". Again, the educator can benefit from getting a quick overview of how students rate the presentation during each lecture.

We want to improve classroom interaction in an easy but effective way. Additionally, the solution has to be free or at least very cheap to allow wide use by educators and students. For this purpose, we have evaluated several relevant approaches by other researchers.

2. RELATED INTERACTION SYSTEMS

The *conversational classroom* addresses interaction in lectures by putting the educator closer to the students (Waite et al. 2003). The educator walks through the aisles during the lessons, asks questions and encourages the students to form groups in which the questions are discussed. However, the lecture is attended by less than 150 students. Applying this approach to courses with more than 500 students is at best difficult.

The *Classroom Presenter* system has been used to facilitate active learning (Simon et al. 2004). Students can fill in blanks in slides or submit their questions or problem solutions. The educator hands out a set of Tablet PCs for student use in each lecture. Student comments appear on the educator's Tablet PC and can be used for discussion or evaluation. Alas, this approach does not scale well to our large lectures due to the cost of Tablet PCs. The potentially large number of submitted slides arriving at the educator's Tablet PC requires a familiarity with active teaching and dynamic adaptation of the lecture to the students' needs. This type of teaching is different from the standard lecture presentation and may not be equally suitable for all educators.

Allowing students to use mobile devices in class can lead to chats and other inappropriate usage. Campbell and Pargas (2003) therefore suggest restricting the use of such devices by a "laptop etiquette". One expectation is that laptops brought to class are always fully charged and in suspend mode. This is only realistic if the campus offers enough freely available power outlets for recharging laptops between lectures. As the university pays for the recharging, and short-outs of the devices may damage university property, universities may be reluctant to allow recharging. Laptop-based solutions thus have to consider the limited availability of power outlets. The following solutions address this by using other mobile devices.

WIL/MA (Scheele et al. 2004) supports providing feedback about the lecture and participating in quizzes. The application requires a Personal Java runtime environment on the mobile device. It is usable on PDAs and Laptops. Evaluation parameters can be configured similar to *TVremote*. As the number of student-owned Pocket PCs is typically very low, the researchers have used project funds to buy 70 Pocket PCs to hand out to students at the start of each lecture. This is clearly not possible for all interested educators.

ClassInHand (2003) offers evaluation, multiple choice quizzes, and submitting text messages. The server can run on a PocketPC which can also control a PowerPoint presentation. Based on web-forms, *SWATT* (Shotsberger and Vetter 2001) allows students to interact with the educator using handheld computers. Answers submitted to the educator's questions can be shown as a dynamically updated bar chart.

The system used by Allert (2003) incorporates interaction and visualizations on PDAs. However, the underlying model of mandatory purchase of a specific PDA by freshmen is not feasible at our university. Adapting the software to the wide range of student devices in use would take too much time.

EduClick (Liu et al. 2003) is based on an infrared receiver and one infrared remote controller for each student. It supports multiple choice tests and states like "lagging behind" and "help request", shown to the educator on an individual basis. The *Classroom Performance System* from eInstruction (2003) is also based on infrared remote controls. It is restricted to 512 users. The direct path of light required by both systems makes them difficult to install in large lecture halls.

All presented approaches depend on the availability of a sufficient number of devices. Special devices, as employed by *EduClick* and the *Classroom Performance System*, are too restricted in their functionality for students to buy them. Therefore, the university instead of the students has to invest money to buy these devices and hand them out.

We have decided to develop a more general approach that consists of a generic open server component and a set of concrete front-ends for different devices. Our system also has to address the university regulations concerning spending funds on buying devices for re-selling or lending to students. For maximum breadth of use at minimum cost to the university, we have decided to support as many different student devices as possible. Our system shall eventually be able to support nearly all types of student-owned mobile devices at no further cost to either the university or the student.

3. THE *TVREMOTE* FRAMEWORK

The *TVremote* framework shown in Figure 1 is conceptually split into a *server* area, an *educator* display and support front-end, and the actual *student interaction tools*.

Supporting a set of user devices for feedback and interaction is best achieved by an open framework. One central design goal for the framework was making interaction as easy and quick as possible. This especially concerns the distraction from the lecture contents occurring when students interact with their devices and the interaction software. We hope that after a short time of use, students will find submitting their interactions as easy and non-distracting as using a standard TV remote control. This expectation is reflected in the framework's name *TVremote*.

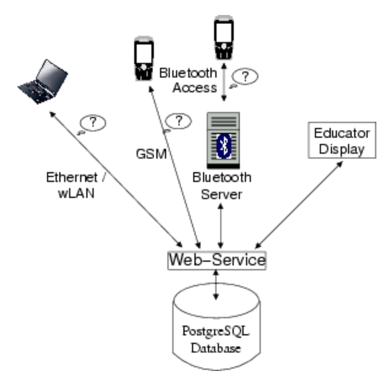

Figure 1. Basic Structure of the *TVremote* Framework

4. *TVREMOTE* SERVER COMPONENTS

The server components consist of an open-source PostgreSQL database accessed by a web-service. Several interaction types are supported: Students can submit messages, which are typed in as free text. Lecturers can define standard templates such as “this slide is too crowded”, which can be selected with the student’s client. Feedback or lecture evaluation is submitted as a rating on a Likert scale. Finally, students can submit their answers to decision polls or multiple choice quizzes. The educator can also publish texts that may be difficult to copy from the projection, such as long numbers or URLs. These can be picked up by the students’ clients. Other types of interaction are conceivable but not currently implemented.

The client adds a timestamp to each submission, which allows i) association with the actual content presented in the lecture; ii) sorting of entries according to their time of arrival to quickly view the “latest question” iii) references to a lecture recording based on the timestamp encoded in the video and the marker set by the student (currently still semi-manual). The system supports three network types: LANs for laptops and PDAs (not considered here) and Bluetooth and GPRS/HSCSD (public carrier) for PDAs and mobile phones. (SMS and WAP were tested experimentally).

Educator display and support: Educators connect via our static (PCs and LCD-tablet) or mobile (Tablet PC based) infrastructure. The display component is responsible for connecting to the web-service, gathering interaction elements, and presenting them in a sensible way to the educator. The current educator display is based on Java. It can be embedded as a plug-in to our educator's control center *Virtual MultiBoard* (Rößling et al. 2004), or it can run stand-alone. Other front-ends have been developed in Visual Basic, Java, and PHP.

A small window is projected in class, containing a numerical value describes the count of current unanswered text messages. The counter increases with each submitted urgent message. This window also gives feedback to the students, as they can see when a new message arrives. By clicking on the button *Fetch new*, the educator can browse the current set of messages. All or some of them can be displayed on a projector by a single click on *Show*. The front-end separates text entries into four categories: *new*, *answered* and *saved* entries, as well as entries suited for a FAQ. Entries can be resorted into a different category by selecting the entry and using the buttons *Saved*, *Answered*, and *In FAQ* under the label *Mark message as*.

The educator's focus should be on the actual lecture and the associated lecture materials. Therefore, an effective support for educators must provide the relevant interaction information “at a glance”. The use of statistical overviews can be very helpful to give the educator feedback without distracting from the actual lecture. For submitted questions, the educator should be informed quickly but unobtrusively of new questions. To avoid abuse, questions should never be displayed automatically, as this can lead to contests between students on “who can submit the funniest entry”.

The answers for multiple choice quizzes or polls are accumulated and evaluated once the educator ends the quiz or poll. A graphical overview clearly shows how many students gave the correct answer and the percentage spread of incorrect answers. The educator can and should use this data to discuss common misconceptions if many answers were incorrect. A short explanation of the reason why a given answer is correct can be very helpful for non-trivial questions. Some students may simply have guessed right, but are likely to be unsure of their answer or the reason the “most likely” answer was correct. Informal polls have also shown that students appreciate a brief discussion of all answers.

5. MOBILE PHONE CLIENT WISH-LIST

A poll taken among computer science freshmen at our university last year showed 54.2% of the participating students owning a laptop, but only 13.3% owning a PDA. In contrast, we estimate that about 90% of our students own another mobile device: a mobile phone. This number is also stated by a German research on youth multimedia and information (MPFS 2004). As we already had a working front-end for laptops (Bär et al. 2005), we decided to develop the next interaction front-end for mobile phones.

Only five per cent of our students stated that they would be willing to pay the fee for GPRS/HSCSD-based mobile phone interaction. A Bluetooth-based solution allows supporting interaction for mobile phones free of charge.

The students’ interface has to be as simple and self-explanatory as possible. There must be a big effort to avoid distraction. If using the application takes too much time, students may lose track of the lecture. The distraction increases with the time students need to submit feedback. Good usability is therefore essential in designing interaction support software. The system should cope for intermittent connectivity. In this case, any communication should be stored locally and synchronized at a later online session.

In our poll, almost half of the questioned students were having problems in asking questions in front of an auditorium of about 600 students. Almost half of the students stated that they would find it easier to ask questions if they could stay anonymous (Bär et al. 2005). Some students tend to exploit anonymity by flooding the system with inappropriate messages. We therefore recommend pseudo-anonymous interaction. This does not show the identity of a student to the educator by default. However, it is possible to resolve a given message's sender if necessary.

6. THE MOBILE PHONE CLIENT

Figure 2 shows the mobile phone client that students use to interact with the educator. The main menu (label 1) directs students to the different interaction types offered. The masks 2, 3, 4, and 5 are used to perform the actual interaction. The educator can configure the number and type of offered interactions. Figure 3 shows interaction offers for questions and comments, evaluating parameters of the lecture and participating in multiple choice quizzes or polls, and retrieving information offered by the educator.

Figure 2. The *TVremote* Prototype running on a Siemens S65 mobile phone

The *Submit text to educator* submenu (label 2) offers a text field where students can enter text. Using the *Options* button of the mobile phone, the student can determine the type of submission. The most common submission type is the urgent question or request to the educator. Here, students hope for a nearly direct answer to their question to better understand the current topic.

Submitting the entry as a general comment is useful for requests that need not be addressed during the lecture (e.g., typos).. The last option in this area is to store the text together with a timestamp (see above for the benefits). The *Speed evaluation* submenu (label 3) allows students to submit their perception of the current lecture on a five-point Likert scale. As stated, educators can configure the parameters available. For the given example, the labels stand for “too slow”, “a bit too slow”, “appropriate”, “a bit too fast”, and “too fast”. The *Quiz* submenu (label 4) is used to participate in multiple choice tests or decision polls. We support capturing the students' attention with the actual question and then providing them with a very simple way to submit their answer. In a *Who Wants to Be a Millionaire* fashion, we have therefore decided to show only the answer labels *A*, *B*, *C*, and *D* on the client. Note that the answers are not necessarily mutually exclusive - students can select any combination of the four possible answers. The answer to the test is again submitted with the *Submit* button. Using the *Retrieve Information* submenu (label 5), students can retrieve information the lecturer offers, such as URLs.

If Bluetooth is saturated or other restrictions lead to the need for an “offline” usage, students can still submit questions, comments, evaluations, and answers to multiple choice quizzes and polls. In offline mode during the lecture, the submitted data is stored locally. That data is transmitted to the server during the next online usage. This is done in the background without any action required by the student. The educator can still answer the questions of the offline users in her office and see the update of the evaluation.

Hardware requirements for mobile phones comprise Bluetooth (or GPRS) support, MIDP 2.0, and the capability to execute about 100 kB of Java code. Such devices are starting to become widespread, market reviews make us expect that at least every second student will own such a device within less than two years.

7. INFORMAL EVALUATION

We could not yet perform a quantitative evaluation of the mobile phone client, but only for the laptop front-end (Bär et al. 2005). A developer's version of the mobile phone client has been in use for a long time, using a working solution based on GPRS without Bluetooth. These initial tests were highly encouraging. The Bluetooth-based solution has been finished only recently. As we plan to use it in a lecture in the summer term, we hope to provide at least anecdotal feedback at the conference - students interest shown is very high.. Additionally, we plan to evaluate the impact of the interaction software and the incorporation of mobile phones on the lecture. Bluetooth is usable across our largest lecture hall (> 800 seats).

Qualitative evaluation using the mobile phone client for *TVremote* proved it to be very easily usable. After some initial work in getting the application installed, the student only has to start it and select one of the interaction types offered for the current lecture. Each of the submenus shown in masks 2-5 behaves as other mobile phone interactions and does not present a problem for today's students.

The educator display ensures that the educator is always kept aware of the number of new interaction elements. The educator has to decide how to deal with the interaction. Some of our users decide to retrieve and answer questions only in the middle and at the end of each lecture. Other educators check new questions shortly after they come in. They then decide on the most appropriate way of addressing them, whether immediately, somewhat later in the lecture, or after the lecture in their office.

As with any new technology, educators should plan in advance how they are going to incorporate it into their lectures. Checking all questions whenever they come up is tempting, but also highly time-consuming, and therefore not suited for all teaching situations. One successful approach involves "staging" multiple choice quizzes or decision polls. The educator announces a quiz with a submission time limit. While the students reflect on the question select their answer, the educator checks recent questions and addresses them after discussing the quiz answers. Other educators use an assistant for screening relevant questions. Our experience indicates that students appreciate quick answers to their questions. However, practically all approaches seem to be well accepted, as long as the educator makes sure to communicate her concrete way of treating questions.

One concern of using the mobile phone-based interaction client is the potential use of the mobile phones for private matters, such as exchanging text messages (SMS) with friends. In large lectures, this cannot easily be prevented. To address this problem regarding notebooks, Campbell and Pargas (2003) recommend a "laptop etiquette" for lectures. Similar rules may also be needed for mobile phone use.

8. CONCLUSION AND OUTLOOK

This paper introduced the *TVremote* framework, which offers interaction during classes by the use of a database and a web-service on a server. The different clients for the teacher and the students connect to the web-service to offer the options as configured. Text messages, evaluation parameters, multiple choice quizzes, and information retrieval are currently supported. The paper focused on the mobile phone client.

Planned extensions include support for mathematical formulae, extension of our live "slide" broadcast / annotation to support small clients, and better automatic referral to contents, concepts, etc.

The server and the client for the educator are not available on a public server, but we are happy to share them with interested researchers who contact the first author by mail. The student clients for notebook and mobile phone are available at *http://www.nu.tu-darmstadt.de/TVremote/*

REFERENCES

Allert, J., 2003. A Companion Technology Approach to CS1: Handheld Computers with Concept Visualization Software. *Proceedings of the 8th Annual ACM SIGCSE / SIGCUE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2003).* Thessaloniki, Greece, pp. 134-138.

Bär, H., Rößling, G., Köbler, S., and Deneke, M., 2005. Evaluation of Digital Interaction Support in a Large Scale Lecture. *Proceedings of the IADIS International Conference on Applied Computing*. Algarve, Portugal, pp. 63-67.

Campbell, A. B. and Pargas, R. P., 2003. Laptops in the Classroom. *Proceedings of the 34th ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2003).* Reno, Nevada, pp. 98-102.

eInstruction Corporation, 2003. Classroom Performance System. WWW: http://www.einstruction.com

Liu, T.-C., Kiang, J.-K., Wang, H.-Y., Chan, T.-W. and Wei, L.-H., 2003. Embedding EduClick in Classroom to Enhance Interaction. *Proceedings of the International Conference on Computers in Education (ICCE)*. Hong Kong, China, pp. 117-125.

MPFS (Medienpädagogischer Forschungsverbund Südwest), 2004. Jugend, Information und (Multi-)Media Studie 2004 (Youth, information and (multi-)media study 2004) WWW: http://www.mpfs.de/studien/jim/jim04.pdf

Rößling, G., Trompler, C., Mühlhäuser, M., Köbler, S., and Wolf, S., 2004. Enhancing Classroom Lectures with Digital Sliding Blackboards. *Proceedings of the 9th Annual ACM SIGCSE / SIGCUE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2004).* Leeds, UK, pp. 218-222.

Scheele, N., Seitz, C., Effelsberg, W., and Wessels, A., 2004. Mobile Devices in Interactive Lectures. *Proceedings of the World Conference on Educational Multimedia, Hypermedia & Telecommunication (ED-MEDIA 2004).* Lugano, Switzerland, pp. 154-161.

Shotsberger, P. G. and Vetter, R., 2001. Teaching and Learning in the Wireless Classroom. *In IEEE Computer*, Vol. 34, No. 2, pp. 110-111.

Simon, B., Anderson, C., and Su, J., 2004. Preliminary experiences with a tablet PC based system to support active learning in computer science courses. *Proceedings of the 9th Annual ACM SIGCSE / SIGCUE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2004).* Leeds, UK, pp. 213-217

Waite, W. M., Jackson, M. H., and Diwan, A., 2003. The Conversational Classroom. *Proceedings of the 34th ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2003).* Reno, Nevada, pp. 127-131.

Wake Forest University, 2003. ClassInHand. WWW: http://classinhand.wfu.edu.