

IMPROVING FEEDBACK AND CLASSROOM INTERACTION USING MOBILE PHONES

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

ABSTRACT

Interaction, such as being able to ask questions, is an important part in lectures. Both educators and students can benefit from software support in very large lectures. 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.

KEYWORDS

Mobile Phone Support, Learner Interaction, Bluetooth.

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. This is the typical scenario for undergraduate computer science lectures at many German universities, where up to 800 students attend a given lecture. While the use of lectures of this size can be debated, various administrative restrictions effectively force us to make the best out of this situation.

In the past, educators have tried various approaches to support interaction in large lectures. The most common interaction form is asking questions during the lecture by raising the hand. However, this is at best difficult to do in such large lectures. On the one hand, the size of the audience effectively prevents the educator from answering the potentially large set of questions. On the other hand, students sitting towards the back are often overlooked. As a compensation for this, students are usually encouraged to come to the educator's or tutor's office during office hours. Additionally, students usually ask questions during lecture breaks as well as after each lecture.

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 lack of central aisles common in large lecture halls also prevents the educator from “getting close” to the students. Some of our students told us about attending such lectures, and mentioned that they try to learn the content without attending. This shows that this approach does not work well for our students.

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.

WILD (Scheele et al. 2004), recently renamed to *WIL/MA*, supports providing feedback about the lecture and participation in quizzes. The application requires a Personal Java runtime environment or compatible on the mobile device. Thus it is usable on notebooks, Pocket PCs, and PDAs. At the beginning of each term, the educator configures the parameters she wants to have evaluated. Students can then submit feedback during the lectures using their mobile devices. As the number of student-owned Pocket PCs is typically very low, the researchers have used project funds to buy 70 Pocket PCs and wireless LAN adapters 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) is very promising, as it incorporates interaction and visualizations on PDAs. However, the underlying model of mandatory purchase of a specific PDA by freshmen is not possible at our university. Adapting the software to the wide range of concrete student devices owned by students would take (too) much time.

EduClick (Liu et al. 2003) is based on an infrared receiver and one infrared remote controller for each student. *EduClick* supports multiple choice tests and states like “lagging behind” and “help request”, which can be 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 some 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 reselling 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

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*.

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*.

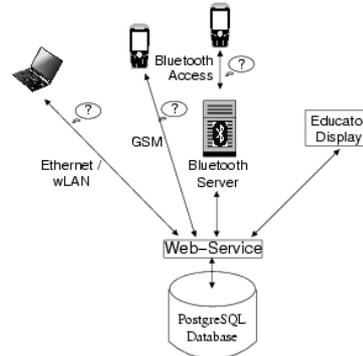


Figure 1. Basic Structure of the *TVremote* Framework

We will first discuss the server and educator components of the *TVremote* framework. The remainder of the paper is dedicated to our latest student interaction client, based on mobile phones.

4. TVREMOTE SERVER COMPONENTS

The server components consist of a PostgreSQL database accessed by a web-service. Both components are based on freely available open source applications. The web-service and the underlying database can handle several interaction types. 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. A coordination of submitted entries with the actual content, such as the slide shown during the submitted question, could also be helpful.

The client adds a timestamp to each submission. Each submission can thus be associated with the actual content presented in the lecture. Additionally, entries can be sorted according to their time of arrival to quickly view the “latest question”. The timestamp also allows us to extend the system to create references to a lecture recording based on the timestamp encoded in the video and the marker set by the student. At the moment, only the appropriate recording file can be opened.

Both educator and students require an appropriate network infrastructure to access the web-service free of charge. For laptops, this is typically based on wireless LAN or Ethernet. PDAs and mobile phones can submit data using either Bluetooth or GPRS/HSCSD. As GPRS/HSCSD is routed by the mobile service provider, this mode of transportation is usually charged with a connection or data transfer fee. PDAs can also use wireless LAN free of charge, if they are equipped with a wireless LAN access card. Alternative data transmissions, such as SMS or WAP, have not been explored in-depth due to the associated data transmission charge.

Figure 1 shows that transmitting the data over Bluetooth requires a Bluetooth server. This server does not have to be a state-of-the-art machine. In our application, a 350 MHz PC is fully adequate to handle the feedback submitted by students.

4.1 Educator Display and Support

The educator needs a computer, usually a laptop, preferably with a private display. Additionally, a network connection is needed to see the questions and feedback during the lecture. The display component is responsible for connecting to the web-service, gathering interaction elements, and presenting them in a sensible way to the educator.

Connecting to the web-service is possible in nearly all programming languages and operating systems. Our current educator display is based on Java and thus also requires a Java runtime environment. Other front-ends for the TVremote framework have been developed in Visual Basic, Java, and PHP.

Figure 2 shows our current educator interface. It can be embedded as a plug-in to our educator's control center *Virtual MultiBoard* (Rößling et al. 2004), or it can run as a stand-alone application.

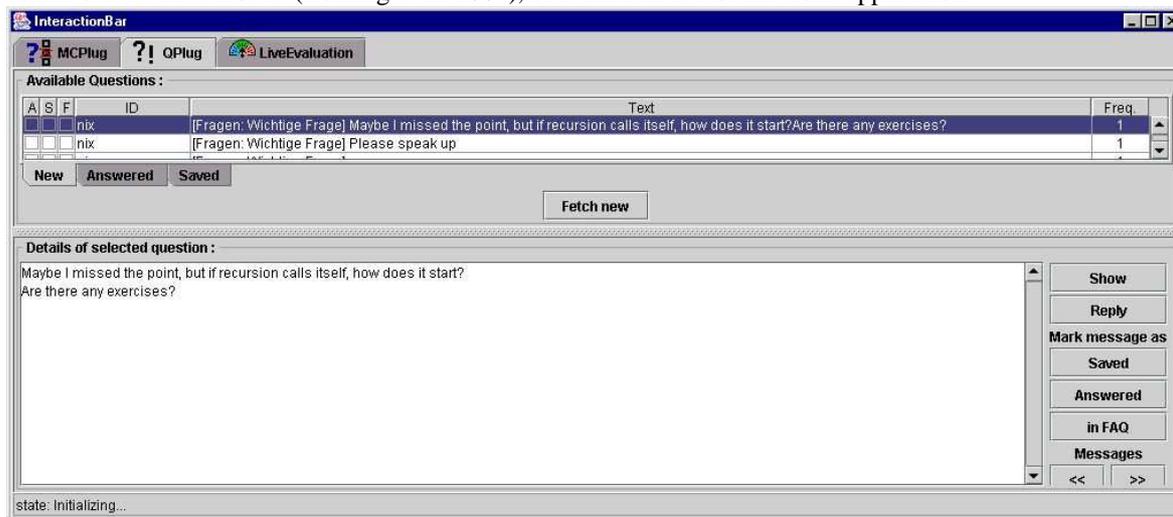


Figure 2. The Educator's Interface

TVremote is most helpful when it is used online. If the students have access to the network, their messages are directly transferred to the educator's client via the web-service. A small window 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*. Navigation is possible using the mouse or cursor keys in the table view at the top, or using the two navigations buttons at the bottom right.

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.

Live communication requires an active network connection. Offline usage is desirable if the network is unreachable. 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. About 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 3 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.

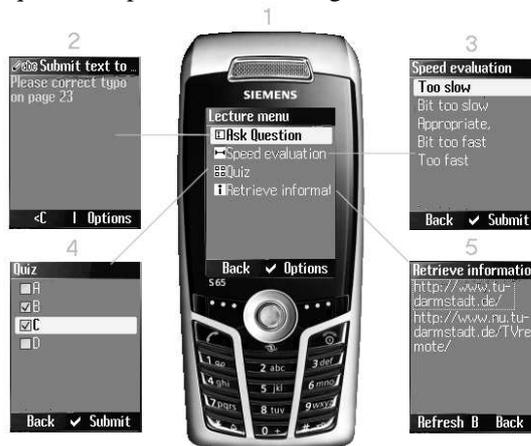


Figure 3. The TVremote Prototype running on a Siemens S65 mobile phone

Submitting the entry as a general comment is useful for requests that need not be addressed during the lecture. For example, this can cover typos on the current slide or general exam questions that do not concern the current lecture. The last option in this area is to store the text together with a timestamp. This allows the student to mark the current section of the lecture as “check this again”. If the lecture is recorded, the student

can later retrieve the lecture recording and navigate to the marked sections to review the topic under discussion.

The *Speed evaluation* submenu (label 3) allows students to submit their perception of the current lecture on a five-point Likert scale. As stated above, educators can configure different or additional parameters they want the students to evaluate. In our lecture, the educator decided to stick with the presentation speed and five steps. The evaluation can be rated by selecting one of the steps. For the given example, the labels stand for “too slow”, “a bit too slow”, “appropriate”, “a bit too fast”, and “too fast”. The descriptions can be defined by the educator. After deciding on one of the settings, the student submits the data with the *Submit* button. The results of the evaluation are shown to the educator in a statistical overview.

The *Quiz* submenu (label 4) is used to participate in multiple choice tests or decision polls. Educators may want to “economize” by not projecting the question and the set of possible answers for the whole audience. In this case, both students without a mobile phone and those with a mobile phone but not “online” are at a disadvantage, or even effectively excluded from reflecting on the question. Transferring the complete question and the possible answer choices to the student's mobile phone makes offline usage impossible. Additionally, it draws the students' attention to their displays to read the question.

We prefer to capture the students' attention with the actual question and then provide 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. This is especially suitable for information that is awkward to copy or easily copied incorrectly, such as long numbers or URLs. It is much easier for the students to copy from a display on their desk than from a projection area, and depending on their device, they may even be able to work with this information using *copy and paste*.

If Bluetooth is saturated or other restrictions lead to the need for an “offline” usage, students can no longer use live interaction. However, they 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.

The mobile phones suitable for a Bluetooth usage of this application need to be equipped with Bluetooth, MIDP 2.0, and the capability to execute Java programs at the size of 100 kB. Such devices are currently starting to become widespread among students. We expect that at least every second student will own such a device within less than two years from now. This is partly due to the relatively low price still being charged for buying a mobile phone in Germany, coupled with moderately expensive connection fees compared to other countries.

There are several ways to install our software on a mobile phone. The student can download the software from the internet using a GPRS or HSCSD connection. The connection fee is about 1€ for each download. Alternatively, most current mobile phones can copy the software from another mobile phone, PDA or laptop, using a Bluetooth or IRDA link. Depending on the mobile phone, this may require additional software, but does not lead to any further costs. Regardless of the software download method, the actual installation procedure takes about one minute, and is at least as easy as installing a given Windows program on a PC.

7. INFORMAL EVALUATION

We could not yet perform a formal evaluation of the mobile phone client. The established laptop front-end of the framework also includes a formal evaluation (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. As this meant paying a GSM fee for each submitted entry, this version was used by only one student. These initial tests have been highly encouraging. We noticed that Bluetooth is usable across our university's largest lecture hall with a capacity of more than 800 students.

The Bluetooth-based solution has been finished only recently. The current solution requires the Students to enter a PIN, which we set to *1234*, so it is easy for students to remember. As we are using this technique in a lecture this summer term, we hope to provide at least anecdotal feedback at the conference. Many students are already interested in mobile phone interaction. Additionally, we plan to evaluate the impact of the interaction software and the incorporation of mobile phones on the lecture.

As can be seen from Figure 3, using the mobile phone client for *TVremote* is very easy. After the quick and easy installation of the application, 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 should therefore 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 and 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. If the educator uses only multiple choice tests, the time slot at which the phones are allowed can be limited.

8. CONCLUSION

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. Possible options are the submission of text messages, the evaluation of parameters of the lecture, multiple choice quizzes, and information retrieval. We have also presented the mobile phone client for submitting interaction elements.

The interaction on the mobile phone can be used over a regular GSM connection at the price charged by the provider. It is also possible to use the interaction over a Bluetooth connection, which does not lead to any costs. As shown in Figure 1, this requires at least one Bluetooth server for each classroom.

In our future work, we plan to include a mask to enter a mathematical equation in *LaTeX* syntax into the user interface and let the server render an image, which can be fetched by the client. This will make it much easier to use such an interaction system in mathematical courses.

We also plan to develop a mechanism to offer the current slide to the students, so they can define the slide their message is related to. More powerful devices such as laptops will then enable students to mark the precise section on the slide that their message refers to.

The server and the client for the educator are currently not available for download. However, 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/>.

ACKNOWLEDGEMENTS

We thank *CellIQ* for lending us different mobile phones for research and development. Siemens also supported us with two brand-new S65 mobile phones. We thank Steffen Czerwinski, Gina Häussge, Jessica Heymann, Elmar Lecher, Marko Mähner, and Marcin Nowicki for their intensive work on different parts of the *TVremote* framework.

Parts of this research were funded by the German Ministry of Education and Research under grant 01NM262.

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>.