SMILE - SMARTPHONES IN LECTURES
Initiating a Smartphone-based Audience Response System as a Student Project

Linus Feiten, Manuel Buehrer, Sebastian Sester and Bernd Becker
Chair for Computer Architecture, University of Freiburg, Georges-Koehler-Allee 51, 79110 Freiburg, Germany

Keywords: Audience Response System, Smartphone, Academic Lecture.

Abstract: In this position paper we are presenting the recent progress in developing our audience response system, SMILE, whose aim is to bring more interactivity into academic lectures. Development started in December 2010 with a programming task force of undergraduate Computing Science students. The constant integration of students as developers is a main concept of SMILE. A first prototype included a multiple-choice quiz module and a slider-based real-time feedback module. The client runs as a native Android-smartphone/tablet app or as a JavaScript web-browser application on any platform. In the winter semester 2011/2012 we have been applying the prototype permanently in a first year Computing Science lecture with approx. 90 attending students. This has been accompanied by a formative evaluation in cooperation with learning psychologists. Final results of the evaluation will be subject to a later publication later. Here we are sharing first tendencies already observable and details of the conception and the ongoing development phases.

1 INTRODUCTION

The idea for an audience response system (ARS) like SMILE emerges from an experience well known to every student in higher education: lectures are often visited by a great number of students; 100 and above. During the lecture all students are sitting in an auditorium listening to a lecturer giving the talk at the black board or in front of the white board with presentation slides. The students remain passive for most of the time unless they get to ask the lecturer a question. This, however, requires the lecturer to give the word to the students which in turn requires the students to make themselves noticeable first.

From our experience students are very reluctant to ask questions in mass lectures. We hypothesised that this is mainly due to the fact that in large audiences it is difficult to judge whether it is just yourself having a question or whether a question is shared by the majority. It is most natural that somebody hesitates to interrupt the flow of a lecture if his question might turn out to be just interesting for himself but not for his 100 or more fellow students. Thus, he might let the moment pass in which his question occurred, which might in turn lead to him not understanding the rest of what the lecturer is putting forward. If the lecturer then stops at a later point to explicitly asks if there are still any questions, our reluctant student is already left so far behind in understanding that he would not even know where to begin his question. Furthermore, he might want to spare himself the moment of being exposed before his peers and the lecturer, if he still does not know whether it is just himself - and maybe the person sitting next to him who he has discussed with - but not the rest of the audience having the question.

Figure 1: The first prototype of SMILE in action.

As a consequence the lecturer too can only guess or must trust his intuition about whether the students understood and approved of the lecture or whether they were merely physically present most of the time and are taking home the frustration of not having understood much while not being noticed.

An obvious way to overcome such drawback

Feiten L., Buehrer M., Sester S. and Becker B..
SMILE - SMARTPHONES IN LECTURES - Initiating a Smartphone-based Audience Response System as a Student Project.
DOI: 10.5220/0003958802880293
In *Proceedings of the 4th International Conference on Computer Supported Education* (CSEDU-2012), pages 288-293
ISBN: 978-989-8565-06-8

would be to decrease the size of classes and get rid of anonymous teaching events like mass lectures, such that a more direct and personal contact could be established both among students and between those and the lecturer. This would naturally be the favourable approach. But decreasing the size of classes would imply to accept less students for higher education or to employ more personnel. Society seems to agree that the former is not desirable, still the latter - which means more capital investment into education - has always been difficult to convey.

Hence, given the situation that mass lectures appear to be a necessary phenomenon in today's higher education, one might go on to think about how to overcome the aforementioned drawback while keeping the lecture setting as it is. In an age of computerisation, where electronic media and communication devices are ubiquitous, the idea for an ARS like SMILE emerges. It envisions a scenario in which all students are equipped with mobile computer devices enabling them to interactively share their current state of understanding among each other and with the lecturer. Further applications to intersperse the periods of passive listening with more activating tasks are thinkable.

Such approaches have been made (Kopf et al., 2005) and taken up by group of first years Computing Science students in December 2010 in a study project involving the programming of smartphones. This was the beginning of the SMILE project. In this position paper we are sharing details from the process of development, what questions we were facing and which decisions we took (Section 2). We describe the functionalities of our first SMILE prototype in Section 3 before giving initial reports about our experiences from an ongoing field test of SMILE in a first year Computing Science lecture with approx. 90 students in Section 4. This field test includes a formative evaluation which was conceived and is carried through in cooperation with learning psychologists. Section 4 includes descriptions about this evaluation as well as first tendencies observable from preliminary evaluation results. Section 5 concludes the paper with a summary and outlines our plans for future developments.

2 DEVELOPMENT

As described in Section 1, the desire for an electronic ARS can emerge naturally from the everyday experience of students. In earlier times German students knew how to anonymously utter their discontent with the lecturer by loudly shuffling their feet under the tables. This custom, however, has today mostly dissipated. Anyway, one would wish for more elaborated ways of giving the lecturer feedback, just as an open-minded lecturer wishes to get constructive feedback from his otherwise mostly anonymous audience.

The problem of sitting in a lecture and not knowing what the lecturer is talking about might be a problem more typical for mathematical and natural scientific lectures than for the humanities, because in the former it seems more likely that the lack of understanding of one concept drastically hampers the understanding of following concepts. This - and the fact that Computing Sciences students and researchers are said to be rather adventurous when it comes to the application of new technologies - might have led to one of the first academic electronic ARSs emerging at a Computing Science faculty (Kopf et al., 2005). The authors developed WIL/MA, which was the main source of inspiration for SMILE. Kopf et al. evaluatied WIL/MA by splitting up a class for one semester giving the same lecture to three groups: one without the ARS and two with differently specified versions of it. Although WIL/MA was well received by the students, the exam results of the three groups did not significantly differ, and eventually the project was discontinued.

In December 2010 a group of first year students at our chair was given the opportunity to do a student project as part of their syllabus in which they should investigate the potentials of applying smartphones and tablet computers in academic teaching. When confronted with the idea of developing an ARS, the group was quickly convinced that their learning experience would benefit from such a system. Thus, the development began and the project was later named SMILE for SMartphones In LEctures. As research staff it was our priority to give the students as much freedom as possible in their design decisions, because we supposed that they were most likely to implement SMILE in a way which would be acceptable from a student's perspective. Furthermore, it turned out that the students' enthusiasm and motivation was at a maximum when they were able to follow their own ideas and initiatives.

We were well aware that from an educational scientific point of view we were merely repeating the experiments of Kopf et al. But their work was done seven years ago when PDAs or programmable cellphones were still a curiosity, whereas today almost every student is equipped with a smartphone or tablet computer. We thus decided to give the idea another chance to prove itself.

To implement SMILE, we first needed a concept for a client/server architecture. The SMILE server would be permanently running on a university computer. As operating system for the virtual machine we took Debian 6 and the SMILE developer students got full remote access with their university user accounts. As programming language for the SMILE server PHP 5 was chosen, executed by the Apache HTTP server and optimised with mod_pagespeed. The database is MySQL 5.

The communication between the SMILE server and the SMILE clients - both student and lecturer clients - was designed to take place over the internet via TCP, exchanging messages including JSON objects. Every communication between the server and a client is securely encrypted through an implementation of the Kerberos[1] protocol. This encryption protocol was chosen because it allows for anonymous logins. We will come back to this difficult issue in Section 4.1.

The SMILE client should be implemented both as a web application to run within a browser on any platform and as a native smartphone app. The web application was programmed with JavaScript using the libraries jQuery and jqPlot. The choice for the smartphone platform was made in favour of Google's Android operating system, as it is not restrictive with installing self-made apps unlike its big competitor iOS, which only allows checked software by registered developers to be installed. To become a registered iOS developer, Apple demands an annual fee. Notice though, that all software used for the development of SMILE is free!

It turned out that the students quickly and mostly autodidactically learned how to find and use the software they needed by means of online tutorials and help forums. The programming of the Android app was done with the Android plug-in[2] for the integrated development environment Eclipse. Eclipse also allowed for the seamless integration of SVN version control via the plug-in subclipse. The SVN repository was hosted on a university server.

The software architecture of both the SMILE client and server was conceived in a modular way, such that a main component - we called the backbone - handled all basic operations like user authentication, session management and securing the communication. The actual ARS functionalities would be handled by extra software modules which would interact with the backbone via a uniform interface. This was done to easily allow the extension of SMILE with new functionalities (modules) without having to change its whole architecture every time. Thus, a lecturer can also customize SMILE for her needs by adding or removing the modules she would like to use.

[1] http://web.mit.edu/kerberos

[2] http://developer.android.com/sdk/eclipse-adt.html

3 FIRST PROTOTYPE

Thanks to the talent and dedication of our SMILE developer students, we had a first prototype ready and running by summer 2011, which we presented to the public at a stall of the university's Science Fair. Its ARS functionalities comprised a real-time feedback module and a multiple-choice quiz module, which are both conceptually adopted from WIL/MA (Kopf et al., 2005).

We also had a forum module programmed, which was inspired by ActiveClass (Ratto et al., 2003). It allows users to post short messages in a forum-like manner. A message is part of a thread whose topic is defined by the user posting its first message. Users can add messages and cast their vote on a thread showing that they find its topic relevant. Such a voting system allows to filter all threads by relevance. The lecturer and students can thus easily see which issue is shared by the majority and should be discussed. We decided though not to promote its use in the field test (Section 4) because we saw a high risk in distraction from the actual lecture and wanted to evaluate the other two modules separately, first. Hence, we will also not explain this module in more detail here. A later version of SMILE will pick up the forum concept again (Section 5).

Real-time Feedback. The real-time feedback module of SMILE gives students the opportunity to make themselves noticeable during the lecture without explicitly interrupting its flow. When the module is selected in a student's client, the graphical user interface (GUI) shows a slider bar with a text label indicating the dimension this slider is rating. The left and right end of the slider are labelled indicating what the extremes of its scale mean. Figure 2 shows such a slider as seen within the native Android client of SMILE. Its label is "Verstaendnis" (understanding) and its extreme positions mean "I lost track!" and "Everything clear!" respectively. The top line shows the name of the current lecture and is part of the Android client's navigation scheme. Figure 3 shows the same slider as it is displayed in the SMILE web client running in a browser window. The tabs above the slider are part of the web client's navigation scheme. The designs of the Android client and the web client differ significantly but their functionalities are equivalent.
The user administration of SMILE allows different user roles. This makes it possible that students and lecturers use the same client software to log in but

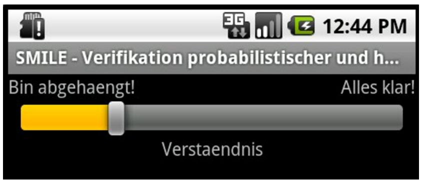

Figure 2: Feedback slider in the students' Android client.

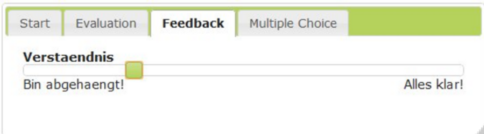

Figure 3: Feedback slider in the students' web client.

get access to different functionalities. Thus, when a lecturer views the real-time feedback module in the web client, she can access the live results of the feedback. These are visualised as a bar diagram as shown in figure 4. Each bar represents the percentage of participating students who have set their slider to the corresponding position. The number of bars can be adjusted. As the students' sliders are stepless, each bar of the result view subsumes the slider positions in its adjacency until the adjacency of a neighbouring bar begins.

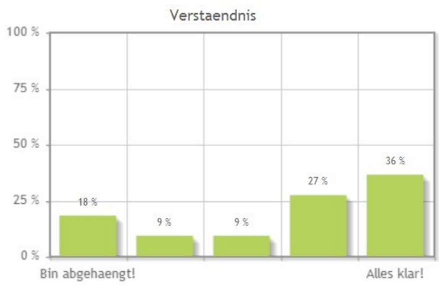

Figure 4: Feedback live results in the lecturer's web client.

A slider for "understanding" is just one possibility. Slider bars can be created and edited arbitrarily by the lecturer. If more than one slider is created, the sliders and results are shown one below the other. We will talk about the possibilities of several sliders in Section 4.1 as well as about the hypothesised effects of the module.

Multiple-choice Quiz. The multiple-choice quiz module is - just as the feedback module - adopted from WIL/MA (Kopf et al., 2005). It enables the lecturer to create a question with different answer possibilities, which she can present to the students at an appropriate time of the lecture. In SMILE a quiz question can either have one possible answer (radio-buttons) or several (check-boxes). Figure 5 and 6 show the students' view of an open question in the Android client and in the web client respectively. As for the feedback module the designs differ, but the functionalities are the same. The shown question has several answer possibilities and asks which of the three given equations (taken from set theory) are true.

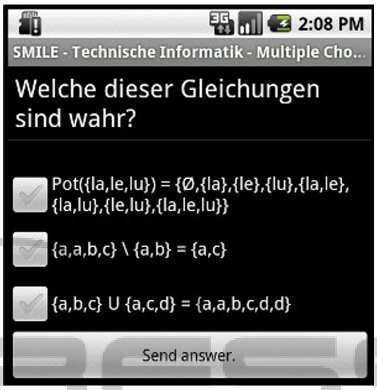

Figure 5: Multiple-choice in the students' Android client.

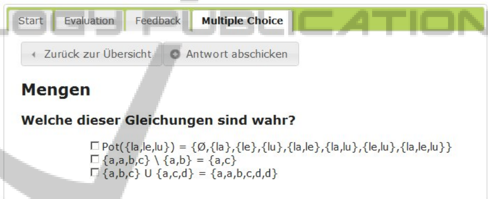

Figure 6: Multiple-choice in the students' web client.

The lecturer can open, close, show or hide questions at any time. If a question is hidden, it cannot be seen by a student user. In order to accept votes for a question it must be opened and not hidden. Every user is only allowed one vote per question and the number of received votes is displayed in the lecturer's web client view. As the lecturer sees fit, she can close the question which leads to no further answers being accepted.

Now, the results can be viewed both on the lecturer's and on the students' clients. Questions with one answer possibility are visualised as a pie chart, questions with several as a bar diagram. Figure 7 shows the results for the above question. It was not too difficult: most students got it right, that only answer 1 and 3 are correct. But three students fell for the little trap of answer 2. A good quiz question always has typical misconceptions as some of its answer possibilities!

4 FIELD TEST AND EVALUATION

When we presented the SMILE prototype at the uni-

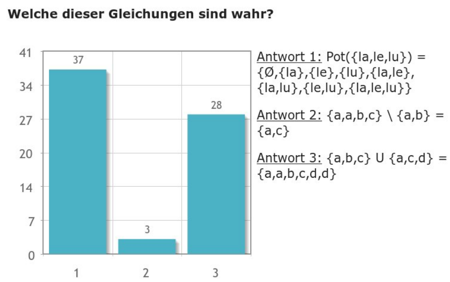

Figure 7: Multiple-choice results in the web client.

versity's Science Fair in summer 2011, reactions were mostly positive. People liked the idea but some where also sceptical about the practicability. Would the ARS not be more a distraction than a benefit? Our answer was that this would have to be empirically evaluated. And so in the winter semester 2011/2012 we were given the chance to utilise SMILE in a first year Computing Science lecture on "Computer Engineering". As the students' attendance is not compulsory for the lecture, the number of participants differs but in average there have been around 90 students in each lecture, which takes place twice a week for two hours.

4.1 Setup

In the first session of the lecture we introduced SMILE to the first semester students with a short presentation and demonstration. This was done by the lecturer, the responsible research assistants but mainly by the SMILE developer students, because we wanted to authentically convey that SMILE is a project by students for students. We also introduced the accompanying evaluation (see Section 4.2).

A difficult issue was how to achieve anonymity within SMILE while at the same time avoid abuse. It is understood that some students like to be anonymous when giving their real-time feedback or multiple-choice answers. Otherwise they might fear SMILE could accumulate their data to see for example which student always puts his slider to "I lost track!". But on the other hand, if we have no user authentication, it would be possible for one person (or malicious script) to log in multiple times and thus tamper with the feedback or quiz results.

As mentioned in Section 2 we implemented the Kerberos encryption protocol because it allows for anonymous user authentication. This is however only possible when the service (SMILE) is not cooperating with another separated authentication server. So it would theoretically be possible to use the university's user account administration for authentication, such that the SMILE server would only know that somebody is authenticated but not who the actual person is. But from the student's point of view he would have to read and understand his SMILE client's source code to confirm that his user data is only sent to the user administration and not to the SMILE server. And even then he would not be sure that the people behind SMILE are not secretly cooperating with the former.

This is still an unresolved question. So for our field test we did the following workaround. We created a fixed number of random user names and corresponding initial passwords and printed them on paper sheets which were handed out randomly to the students in the lecture. The password could later be changed by the students. Thus, each student could only login once, but it could not be known which person is behind which user name.

*

Our hypothesis was that students will feel more encouraged to ask questions in the lecture when they know that is it not just themselves having difficulties. Thus, it would be necessary to let them see the lecturer's view of the live-feedback results all the time. This is achieved by running the lecturer's SMILE web client in the browser of an extra laptop computer whose screen is projected on a separate big screen next to the actual white board, as seen on the photo in figure 1. The screen of the extra computer is facing the lecturer such that he can observe the live results too while facing the students.

Kopf et al. had two feedback sliders for their experiments with WIL/MA: one for the lecture's complexity (too high – too easy) and one for its speed (too fast – too slow). We decided to only use one slider for understanding ("I lost track!" – "Everything clear!") as described in Section 3. Thus, our slider's desired position is on the far right; unlike those used in WIL/MA whose desired positions were in the middle.

The feedback live results are visible for the whole lecture on the extra screen. When a multiple-choice quiz happens, the lecturer switches the view in his SMILE web client so that the students see it as well on the extra screen. The question and answer possibilities are read aloud followed by a period of 2-5 minutes during which the students can send their answers. Then, the results are viewed and discussed before the lecture continues and the extra screen goes back to showing the feedback live results. On average we have 1-2 quizzes per 2 hour lecture.

4.2 Evaluation

In summer 2011 we showed our first prototype to learning psychologist, who agreed to assist us in a formative evaluation accompanying the planned field utilisation of SMILE. Our main aim was to find out if students experience SMILE as a benefit or not. An electronic questionnaire with 22 items was created to assess possible developments in the students opinion. We implemented the questionnaire as an additional SMILE module called "Evaluation", which allows students to answer the questionnaire once between two consecutive lecture sessions.

Most items are questions which can be answered via a 5-step slider ranging from "agree" to "not agree". Abstaining from answering a question is possible via a corresponding check-box which is activated by default and automatically deactivated when the slider is moved. A more detailed description of the evaluation setup and results will be subject to a later publication. For now we can share the following observations:

Students seem to appreciate the multiple-choice module more than the real-time feedback and would wish that there are more than just 1-2 quizzes per lecture. The number of students participating in the quizzes is approx. 50% of the attending students. The feedback is only constantly used by approx. 15%. This seems primarily due to technical difficulties we have had with the wireless LAN in the lecture hall. Glitches in the students' internet connections lead to frequent automatic logouts making the constant usage of the feedback module a nuisance. Furthermore, many students have the security functionalities of their smartphones enabled which requires them to enter a password or gesture each time after the device has been shortly inactive.

We also realised that the percentage of students taking part in the evaluation questionnaire regularly has been low. We therefore conducted another evaluation via paper questionnaire with items aiming at these issues as well. The results will be subject to the later publication explaining the whole field testing period in detail.

5 CONCLUSIONS AND FUTURE DEVELOPMENTS

We were able to show that it is possible for a group of talented undergraduate students to autonomously implement and maintain a complex system like SMILE using university hardware and contemporary free software and online resources. The development is in ongoing progress. At the moment we are in discussion with a group of new students, who have taken part in the "Computer Engineering" lecture where SMILE is tested and want to join the development team. With some of the former SMILE developers leaving the team, SMILE might thus be passed down from year to year and from students to students.

A next development would be to get rid of the extra screen in the lecture hall and transform the lecturer's SMILE client into an expandable toolbar below the whiteboard slides. Furthermore, we are taking up the forum idea again, thus implementing a backchannel as recently suggested by (Gehlen-Baum et al., 2011). Aiming at devices with larger screens (i.e. tablet or laptop computers), forum entries could be associated to positions in the lecture slides as suggested by (Anderson et al., 2003).

Beside various modifications to improve the usability of SMILE, we are particularly thinking about integrating the multiple-choice quizzes more seamlessly into the lecture slides. A perfect way to do this has recently been demonstrated by Thrun and Norvig in their online *ai-class.com* with several thousand participants worldwide. Concepts like *ai-class.com* might eventually more and more replace the classical lecture model. With a project like SMILE we hope to make it still worthwhile for larger crowds of students and lecturers to physically meet from time to time.

REFERENCES

Anderson, R. J., Anderson, R., VanDeGrift, T., Wolfman, S. A., and Yasuhara, K. (2003). Promoting interaction in large classes with computer-mediated feedback. In *Proceedings of CSCL'03: the International Conference on Computer Support for Collaborative Learning*, pages 119–123, Bergen, Norway. Kluwer.

Gehlen-Baum, V., Pohl, A., and Bry, F. (2011). Assessing Backstage–A Backchannel for Collaborative Learning in Large Classes. In *Proceedings of the 14th International Conference on Interactive Collaborative Learning (ICL 2011)*, pages 154–160, Piestany, Slovakia.

Kopf, S., Scheele, N., and Effelsberg, W. (2005). The interactive lecture: Teaching and learning technologies for large classrooms. Technical Report TR-05-001, Mathematics and Computer Science Department, University of Mannheim.

Ratto, M., Shapiro, R. B., Truong, T. M., and Griswold, W. (2003). The activeclass project: Experiments in encouraging classroom participation. *Computer Support for Collaborative Learning 2003*.