Tweedback: A Live Feedback System for Large Audiences

Jonas Vetterick, Martin Garbe and Clemens Cap
Department of Computer Science, University of Rostock, Rostock, Germany

Keywords: Feedback, Live Feedback, Lecture, Peer Instruction, Chatwall, Web Application.

Abstract: Live feedback systems have been proven to be suitable for teachers' needs. Especially in classes with a lot of participants feedback in an electronic form can create additional value by getting feedback from all students. The live feedback system Tweedback incorporates three feedback possibilities. One form of feedback is Peer Instructions to test people's knowledge using multiple choice questions. Additionally the possibility for students to rate the teachers' Speech Parameters could improve the lecturers course value. Furthermore the ability to ask questions anonymously could lower the threshold for students and may result in a better understanding.

1 INTRODUCTION

Nowadays students often use mobile devices, such as smartphones or tablets, during a lecture to communicate with each other. This offers an opportunity to use mobile devices as a new communication channel between a teacher and her students.

Often teachers want to check students' understanding of a previously explained issue, for example the functionality of a complex concept in computer science or the diagnose of a patient's symptoms in medicine. Asking a small group of students is feasible, but asking a larger group can be very inconvenient, where this may lead to turmoil in class and will often be focused only on a small number of participants. There are solutions, which utilize a remote control to allow the students to choose predefined answers (for example "Qwizdon Q5" (Qwizdom Ltd, 2012), "Powervote" (Powervote, 2013) or "voting4life" (Art4live GmbH, 2013)). This simplifies this proposal to ask a large group of students. Imagine a medical course, where the lecturer describes the symptoms of a disease. The lecturer then frames four possible diseases as answers and each student has to vote for one.

Solutions for this kind of feedback, which use an extra device, are on the one hand very comfortable. The usage is simple for both, students and lecturers, so there is no need for an extra explanation. On the other hand there are a lot of disadvantages. All devices have to be maintained by replacing empty batteries, keeping them clean, retaining them and ensuring they are complete in quantity and quality.

To avoid these disadvantages, another possible solution is to use the mobile devices, the students own already. By using their own devices, a lecturer could ask a question to a large group of students without providing extra devices.

All in all there are three different types of feedback, which will be explained in the following. All three types have the possibility in common to get live feedback. Live feedback means the ability of getting instant feedback from a large group of people in a classroom scenario. Our goal is the implementation of a tool, that allows teachers to use all three types of live feedback in their lectures, without having to care about technical details.

The previously presented scenario describes the Peer Instruction concept (PI), where "the instructor presents students with a qualitative (usually multiple choice) question that is carefully constructed to engage student difficulties with fundamental concepts. After that the students consider the problem on their own and contribute their answers in a way that the fraction of the class giving each answer can be determined and reported." (Redish, 2005)

The second type of live feedback is the Chatwall (CW), also known from facebook.com and twitter.com. Students have the possibility to ask questions and are able to vote for other students' questions to make them more important. Then the lecturer can use this information to clarify obstacles. This concept is very similar to the backchannel idea, first mentioned by the research group of Franois Bry. (Gehlen-Baum et al., 2012)

Whereas the Peer Instruction concept and the

Vetterick J., Garbe M. and Cap C..
Tweedback: A Live Feedback System for Large Audiences.
DOI: 10.5220/0004414501940198
In *Proceedings of the 5th International Conference on Computer Supported Education* (CSEDU-2013), pages 194-198
ISBN: 978-989-8565-53-2

Chatwall/Backchannel have been well-known for several years, the third kind of live-feedback is not yet popular and has not yet been fully investigated. The main idea is that the students are able to mark a single Speech Parameter as inconvenient. The talking speed is for example a possible Speech Parameter. By giving the audience the ability to mark the talking speed as too fast, the lecturer can reduce her talking speed to improve the comprehensibility.

All three kinds of live feedback are presented in detail in the following chapter. After that we define the functional and nonfunctional requirements of a general live feedback system. In the following we present our architecture and illustrate design decisions we made for Tweedback to fulfill these requirements. The last chapter summarizes all the work we have done and outlines our future work.

2 STATE OF THE ART

As previously mentioned, there are three kinds of live feedback. This chapter overviews existing solutions, which either implement one kind of live feedback or a composition of two or more. We introduce first the implementations of the Peer Instruction concept and then a solution for giving feedback based on a lecturer's Speech Parameters.

The University of Paderborn realized the Peer Instruction concept as a closed-source web application called PINGO (Reinhardt et al., 2012). They built an application that is accessible on any device, mobile or not. Using this application a lecturer, who must have an account at the University of Paderborn, can add surveys during the lecture. A survey thereby exists of a multiple choice question. The participants, mostly students, do not need to have an account and answer anonymously to the lecturer's questions. It is not possible to add free text questions and the source code is not open.

The Peer Instruction concept is also realized as a prototypical web application (University of Rostock, 2012) at the University of Rostock. Thereby the focus is rather on the usability and performance than the stability. So this application has a minimalistic user interface and thus provides maximum compatibility with mobile devices.

One solution, that also handles just a single kind of live feedback has been developed at the University of Freiberg and is called myTU (Technische Universität Bergakademie Freiberg, 2012). It is available for the mobile operating systems Android and iOS. It is more a manager for student's life than instrument for live feedback, but students can use it to rate the speed of lecturer's speech.

Combining a Chatwall and the ability to rate the lecturer's Speech Parameter is the idea behind the web application Backstage (Gehlen-Baum et al., 2012), developed and researched by F. Bry. Backstage was designed to investigate the use and the design of a digital backchannel by implementing a prototype that looks and acts like twitter, but is used for a lecture. The students may ask questions to the whole audience, including the lecturer. The lecturer also may ask questions to the students using Backstage. Backstage is focused on large classes, but is used only with a small number of students for the experimental studies (approx. 20 students). Due to its design using a synchronous Java web server as Jetty, it may be slow under heavy use with multiple large lectures (more than 100 participants) in parallel.

A project, which is most similar to our approach is called SMILE 2.0 (Feiten, 2012). It has been developed by the University of Freiburg and allows students to ask free questions and rate the lecturer's Speech Parameters using a social platform. Furthermore it has the functionality of a Peer Instruction application. Unfortunately it is not open, neither to use nor the source code, so we could not evaluate it.

Our goal is a composition of different types of live feedback, namely PI, CW and SP.

3 AIMS AND REQUIREMENTS OF TWEEDBACK

The major aim of our system is to enable feedback for students and give lecturers the possibility to react on the feedback in time. Especially in large classes verbal one to one communication with many people is not feasible. Here feedback systems can assist the teacher.

3.1 Functional Requirements of Live Feedback System

Feedback can be given in different ways. We concentrate on three categories. The first functional requirement is Peer Instruction (PI). Teachers should be able to start surveys ad-hoc in lecture or start prepared ones. A survey is implemented as a multiple choice question. Results of these surveys should be accessible for the lecturer at all time.

Besides Peer Instruction, students have additional possibilities to give feedback regarding the content of course and form of presentation. Speech Parameters (SP) can be rated throughout the lecture. Example categories are speed and understanding. A threshold regulates how many votes are needed in a given period of time to notify the lecturer. The teacher will be notified on a small screen about the current situation.

The third functional requirement, called Chatwall (CW), is feedback in form of asking particular questions, rating and answering them. For lecturers it is hard to give a presentation and read questions from the audience at the same time. Therefore an unfiltered forwarding of audience questions will not be convenient. The system gives the audience the possibility to ask questions on a Chatwall-like platform. At the beginning questions are proposed and displayed only on students' devices. After questions have been proposed others can vote for these. This can result in high rated questions concerned by many others. A threshold decides which question will be presented to the lecturer. A log of the Chatwall can also be used by the lecturer afterwards to evaluate his presentation.

3.2 Non-functional Requirements

Besides the functional features as described above, Tweedback has many other requirements which aim best possible user experience. First of all, interaction with the system has to be fast and responsive. Waiting while interacting with the application is decreasing acceptance of users. A second requirement with direct impact on acceptance rate is an intuitive and minimalistic user interface, that even fits on very small mobile devices. Tweedback will be used by many laypersons. According to (N)ONLINER Atlas 2012 (Initiative D21 e.V., 2012), which represents the situation in Germany, 96.9% of young people (20–29 years) have Internet experience. So we can assume that they are able to use even complicated websites. According to the lecturer's view, it is assumed that teachers have less experience with the Internet (e.g. 60–69 years, 60.4%). We do not know what people will use our system in the future, how old they are or what preferences they have. So we cannot suppose that teachers are familiar with complex websites and their navigation features. According to the students' side it can be assumed that there is more experience in dealing with complex websites due to the age of students.

Students use their mobile devices to access Tweedback. It is not possible to control which device will be used. This is a design decision. The advantage is that users maintain their own devices. The disadvantage is that the interface of a live feedback system needs to be as flexible as possible on students' devices.

With a growing number of users the system needs to be scalable. This is especially a matter of concern when many surveys start at the same time. In these cases many students take part at a survey simultaneously. This causes an access peak. Situations with thousands of connections need to be handled.

For users we want to make the first step of participation in Tweedback as easy as possible, which leads to an anonymous access for students. We do not want users to pass an initial account creation procedure. We believe in success of open systems.

To generate additional value compared to "offline" feedback, users can not be blamed for wrong answers. It is a common phenomenon that people do not participate in surveys because they are afraid to say wrong answers in the front of others. But these people are able to participate in a feedback system when users are anonymous. This is the second reason why anonymous access is preferred.

Finally the system has to be modular to easily implement new forms of audience interaction in the future.

4 ARCHITECTURE OF TWEEDBACK

In order to provide an application for many mobile devices, it is sensible to build a web application that simply serves HTML. Because most mobile devices are able to render HTML5, Javascript and CSS, there is no need to develop a native app for all the mobile operating systems (namely the most common used: Android, iOS and Windows Phone) (Kamboj and Gupta, 2012). By using just one browser-related client interface, we do not have to manage and maintain multiple different platform variants. Even if there are frameworks that promise to handle these issues, there would always be multiple different code repositories. This would unnecessarily increase the development effort and support.

Furthermore the user interface shipped by the web application should not only fit on most kinds of screens (for example on a mobile device and/or a notebook), but it should be easy to use, too. While the first can be assumed by using a well-known framework for web user interfaces, the last is hard to achieve. Our approach is to use the bootstrap framework from twitter. It provides a set of common user interface elements, such as a header bar or a button, and organizes the way it is visually structured. Additionally we want to subdivide the user interface into different views, in such a way that each feature has its own view. Offering the lecturer a choice of features, students are able to see which feature is enabled by recognizing which view is currently activated. A feature implements one of the previously explained functional requirements, namely the Peer Instruction, the Chatwall and Speech Parameter.

To implement anonymousness there has to be a concept to ensure that a lecturer can not trace back the students id. Technically we solve this issue by allowing everybody to participate. Nobody has to register herself or login with her university account. Another fact arises with the ability to be anonymous. It is maybe possible to manipulate the system by overusing a function or trying to distract the lecturer. We prevent such manipulations by defining certain thresholds and time locks. A student for example can not ask multiple questions at a short time ($<$1 second), so we disable her functionality to ask questions right after a question is asked.

Responsiveness and scalability are difficult to achieve, so we need sophisticated technologies to solve these requirements. Fortunately both can be handled at once by using an asynchronous web server working with two-way communication channel in tandem. Investigating in numerous solutions for asynchronous web servers, we decided to use the tornado web server, developed by facebook. It is event-based, well-known for stability and very good performance. It is possible to handle more than 5000 connections simultaneously on commodity hardware. A two-way communication channel is necessary, because both, the server and the client, have to send messages to each other. The client hereby is the browser, of the lecturer as well as of the students. Imagine for example the use of the Chatwall, where one student asks a question. To view this message to all the other students, it has to be delivered to their browsers. Here the student's browser sends the question to the server. Then the server has to push this message to all the other students (and lecturers maybe, too) browsers. Without the use of a two-way communication this would not be possible.

We plan to use the library socketIO as the two way communication layer. It makes use of WebSockets, defined in HTML5. Furthermore, it is well implemented for Javascript and for the tornado web server. Additionally, it has several fallbacks for browsers not supporting Websockets. These fallbacks make use of AJAX to emulate a two-way communication over HTTP 1.1.

Figure 1 summarizes the technology stack previously presented. All permanent data will be stored in a non-relational database, because there is a flexible scheme necessary to develop this web application.

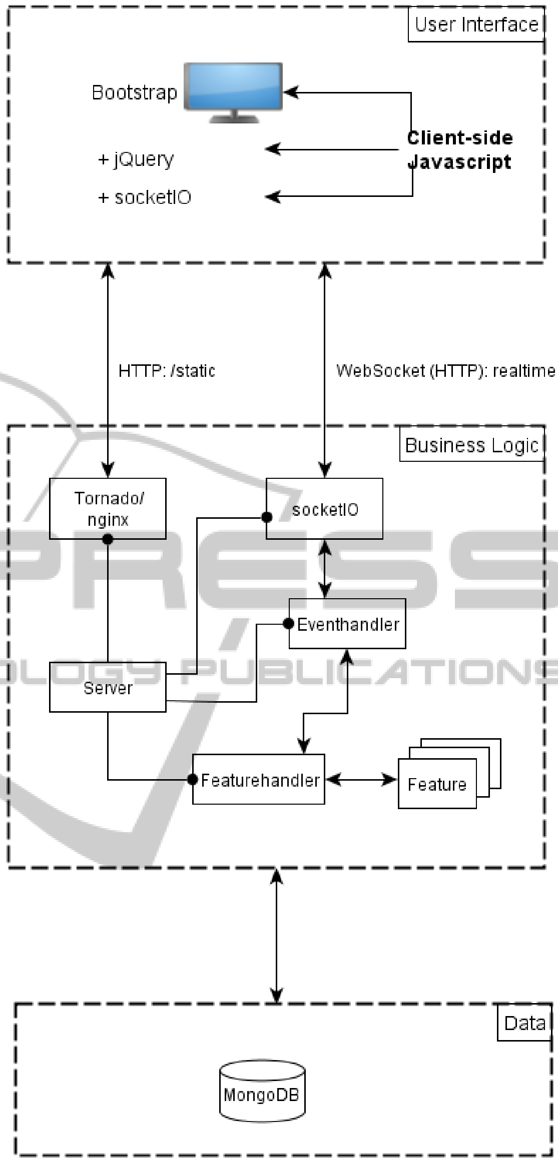

Figure 1: Technology Stack.

5 CONCLUSIONS

Many people have mobile devices for communicating with others or simple Internet browsing. These devices can be used to give feedback within large classes. The presented live feedback system Tweedback uses these mobile devices. People can give feedback in one of three ways: Peer Instruction, Chatwall or rating Speech Parameters. Besides these functional requirements the feedback system has to be fast and responsive, easy to use by laypersons, compatible to many mobile devices and scalable. These requirements are accomplished with user interface frameworks for mobile web which assure compatibility to most user devices. Event-based asynchronous communication is used for scalability and anonymous access is allowed to assure an easy participation and to give users the chance to give feedback which they would have refused in front of many other people.

REFERENCES

Art4live GmbH (2013). voting4live: http://www.voting4live.de/ [23.01.2013].

Feiten, L., . B. B. (2012). Smile smartphones in lectures: Initiating a smartphone-based audience response system as a student project. 4th International Conference on Computer Supported Education (CSEDU 2012).

Gehlen-Baum, V., Pohl, A., Weinberger, A., and Bry, F. (2012). Backstage designing a backchannel for large lectures. In Ravenscroft, A., Lindstaedt, S., Kloos, C., and Hernndez-Leo, D., editors, *21st Century Learning for 21st Century Skills*, volume 7563 of *Lecture Notes in Computer Science*, pages 459–464. Springer Berlin Heidelberg.

Initiative D21 e.V. (2012). *(N)ONLINER Atlas 2012. Eine Topographie des digitalen Grabens durch Deutschland.* TNS Infratest, Mnchen.

Kamboj, V. and Gupta, H. (2012). Mobile operating systems. *IJEIR*, 1(2):115–120.

Powervote (2013). Powervote: http://www.powervote.com [23.01.2013].

Qwizdom Ltd (2012). Qwizdom q5: http://www.qwizdom.co.uk/q5.php [23.01.2013].

Redish, E. F. (2005). Peer instruction problems: Introduction to the method.

Reinhardt, W., Sievers, M., Magenheim, J., Kundisch, D., Herrmann, P., Beutner, M., and Zoyke, A. (2012). Pingo: Peer instruction for very large groups. In Ravenscroft, A., Lindstaedt, S., Kloos, C., and Hernndez-Leo, D., editors, *21st Century Learning for 21st Century Skills*, volume 7563 of *Lecture Notes in Computer Science*, pages 507–512. Springer Berlin Heidelberg.

Technische Universität Bergakademie Freiberg (2012). mytu app: http://mytu.tu-freiberg.de/ [22.02.2013].

University of Rostock (2012). Wwm-hro: http://wwm-hro.tk [22.01.2013].