This is the peer reviewed version of the following article: Knutas, A., Hämäläinen, H., Ikonen, J. and Porras, J. (2013), Implementation of an embedded mobile device based feedback system for real-time audience feedback. Comput. Appl. Eng. Educ., 21: E31–E40. doi: 10.1002/cae.20570, which has been published in final form at http://dx.doi.org/10.1002/cae.20570. This article may be used for non-commercial purposes in accordance With Wiley Terms and Conditions for self-archiving.

Implementation of an embedded mobile device based feedback system for real-time audience feedback

Antti Knutas, Harri Hämäläinen, Jouni Ikonen, Jari Porras

Lappeenranta University of Technology

P.O. Box 20, FIN-53850 Lappeenranta, Finland

firstname.lastname@lut.fi

Abstract: The paper presents design and implementation of a mobile device based feedback system for lectures. The system permits participants to use mobile devices to post questions to the presenter's screen without interrupting the spoken flow of the presentation. Tests in conferences showed that the system increases the amount of feedback.

Keywords: learning, web, mobile, wireless, embedded

1. Introduction

Conferences and the methods of learning used therein have not changed greatly despite the increasing availability of electronic tools. The most common approach to using new electronic tools has mostly been to imitate older tools, like computer slideshow software replacing overhead projectors, without considering how to use the new tools to change the learning environment. Communication in academic conferences is still mainly one-directional [1]. The current practice is that the presenter gives a presentation about his topic, followed by a limited amount of time for questions and comments from the audience. This approach to conference presentations has an obvious drawback: Only the most communicative and active members of the audience present their questions or comments. Additionally, there is no written record of the event, which hinders reference to and utilization of the feedback after the event.

Similar circumstances also occur within the context of classroom teaching when class sizes become large, as the case is with several university classes [2]. Typical university lectures are one-directional events in which the teacher talks and the students mainly listen. Even when teachers try to increase interaction with students by encouraging them to ask questions, some students are too shy to ask questions or express their opinions. Students may feel uncertain of their knowledge and be afraid of taking the risk of embarrassing themselves by posing inappropriate questions [3]. An anonymous method or system for presenting questions and comments has potential to increase the degree of interaction in university lectures.

Based on experiences of feedback systems in learning environments [4,5], it was decided to investigate how a similar approach could be utilized in more formal conference events. The solution adopted was the implementation of a mobile device based system that allows members of the audience to write feedback during the lecture rather than expressing it aloud. This has the benefit that feedback can be given without interrupting the speaker, breaking the flow of the presentation, or disturbing other members in the audience. The feedback is shared immediately both online and projected on the screen to the whole audience, which gives them a possibility to process the feedback. To the speaker, it gives an option to either respond to the comment immediately or at a later time, e.g. during the open discussion after the presentation. Having at least a part of the questions and feedback stored in digital form permits access to these comments at a later time.

Not much research into conference interaction systems exists, but online classroom interactions have been under research for over a decade at the time of writing. The assumption, supported by previous research [2,6], is that the added interactivity will increase user interest during the presentation itself, encouraging participants to give more feedback and giving more topics for discussion after the presentation itself. When designing our solution for an online interaction system, lessons learned from classroom interaction systems were applied [2,7] and the special requirements of a conference environment taken into account: The *educational goal* of the solution is *increased opportunities for interactivity*, which in turn will lead to increased user interest and more feedback to the lecturer.

This paper describes a system which can be used in lecture environments to send and receive feedback in real time, displaying the feedback both to the lecturer and the members of the audience. Feedback in this context is 140 to 2000 words long written messages aimed to the lecture. The system is web-based and is integrated into the conference presentation environment with a large public screen that shows feedback or questions from the audience. No dedicated client machines are used because it is assumed that increasingly audience members will have some kind of web-enabled mobile device with them.

The rest of the paper is organized as follows. This introduction is followed by a description of the use of mobile solutions for interactive learning. Section 3 presents the implemented system and its design. Section 4 reviews the system installed on an embedded device and describes its functionality. Section 5 concludes the paper.

2. Mobile interaction in learning

There has been a considerable amount of research on interactive electronic systems that support classroom communication in the form of quizzes and feedback [8]. In general, classroom-based interactive electronic systems have been seen to have positive results, with the students feeling that the interactivity provided by the systems enhanced lectures [2,9,10]. The students also felt that the added interactivity increased their interest in participating in classes. The use of electronic devices does not necessarily lessen interaction between the students, either. When designed into the lecture, the use of classroom interaction systems encourages collaborative learning [11]. Overall, the new generation of students has been seen to be more comfortable using digital devices in personal life and while learning [6].

In addition to being used as teaching aids, online interaction tools have been used for computerized tests or quizzes to gain more accurate understanding of how well the class has understood the course material [12,13]. Despite the different tools used to measure student understanding, the test results are similar to what pen and paper tests would achieve after the student have become familiar with the testing devices [13,14].

Studies have shown that because classroom interaction systems increase learning and attentivity, they lead to positive learning outcomes compared to similar classes not using such systems [2,12]. However, no research has considered the use of interaction systems to improve conferences, despite conference presentation environments and school lecturing environments being similar. The positive results from classroom situations suggest that increased electronic interaction could also benefit conference environments.

While adding interactivity is beneficial from the viewpoint of the audience, adding interactivity to lectures might bring additional challenges to speakers, despite the attempt not to change the basic nature of the event. Simpson and Oliver [10] in their review of the pedagogical use of electronic voting systems noticed that the effectiveness of the systems depended partly on how well the lecture was designed to utilize the learning tools. It is possible that the full benefits of a feedback system might not be gained if the lectures are not planned with instant feedback in mind. However, in some cases the material designed for the online system can be reused in other lectures of the similar type, which eases the workload after the initial effort [11].

Several kinds of classroom learning systems with mobile interfaces are available. According to Rochelle [15] the simplest and most common system until 2003 was a polling system where the lecturer can arrange a quick but simple quiz and review the answers. Most of these response systems are completely controlled by the teacher and require the teacher to initiate all interaction by presenting a question and providing a list of answers. One first generation example is the Classtalk polling system with a dedicated hand-held, wired device containing a row of buttons for multiple-choice answers [9].

Second generation systems like Numina II [16] differ from first generation systems in three major ways: More complex quizzes can be presented to the audience, the systems have more advanced hardware, utilizing computers or handheld devices instead of screenless machines, and cable links are replaced with wireless ones. Numina II is designed to run on Windows Mobile PDAs which feature a color touch screen and browser access to Numina's web-based interface. All interactions are still initiated by the lecturer and the available answers are still limited to multiple choices. Another polling system, the Open Client Lecture Interaction System (OPCI) [4] has similar but slightly simpler multiple-choice answer features. Additionally, in OPCI students can send SMS (Short Message Service) messages as feedback and give evaluations of lectures.

Third generation systems have more features and diverge more in functionality to address specific problems, without completely replacing the multiple answer type of programs. For example, the Classroom Presenter system [17] is a freeform slideshow and drawing program based on Tablet PCs. The lecturer presents a problem in the form of a slide and students can make a local copy on their tablets. Students can answer by drawing on the slide and submitting it back to the lecturer. The lecturer can review all the submissions and pick one or several to display on the lecture room projector. Another recently developed system is the MLI system (Mobile Learning Interaction) [18] in which students can submit questions to the lecturer using a client running on a smartphone and vote on questions submitted by other students. If a question collects enough votes, the lecturer is alerted by the system. The ITSE (Interactive Teaching Support Environment) [5] application takes a different approach. The ITSE application offers a web-based interface for mobile clients and the server software has a version which runs on an embedded device platform. In the application all feedback is anonymous, equal and instantly projected to the screen in contrast with MLI's queued and evaluated feedback approach.

Webinar, or virtual classroom learning tools, can also in theory be used in local classrooms. However, they require more technical infrastructure and can be difficult to manage [19,20]. Additionally, webinar software is designed for distance learning, which is a different method of learning than lecture interaction. One common hindrance in distance learning is lessened peer interaction [21].

In short, there are a wide variety of classroom and lecture interaction systems. Some newer generation systems improve on the quiz mechanism introduced in the first system. Other newer systems branch out in ideas and introduce new kind of concepts for interaction. The Table 1 presents a comparison between the available features in different systems.

Table 1. Learning tool comparison chart

	Classtalk	Numina II	OPCI	Classroom Presenter	MLI	ITSE
Wireless network support		WLAN	WLAN	WLAN, cellular	WLAN, cellular	WLAN, cellular
Client device	Proprietary	Proprietary tablet	Any smart-phone or equivalent	Proprietary tablet	Smart-phone	Any smart-phone or equivalent
Communication initiated by	Lecturer	Lecturer	Both	Lecturer	Audience	Both
Communication	Linear, realtime	Linear, realtime	Bi-directional, realtime	Linear	Non-linear	Non-linear, realtime
Interaction features	Quiz	Quiz	Quiz, lecture feedback	Quiz with graphical answers	Lecture feedback	Lecture feedback
Requires system-specific teaching material	Yes	Yes	Yes	Yes	No	No

Based on the comparative evaluation of the systems presented earlier, most of them require specially prepared material for the lecture, which can discourage lecturers from trying out the system. Half of the systems also feature linear interaction in the sense that there must be specific time reserved for the use of the system in the lecture, during which no other teaching can be done. There is one niche that other systems still do not fill: A need for a system which requires little setup, very little design on the part of the lecture material, but still provides added opportunities for interaction during the lecture itself.

3. System design of a mobile feedback system

The main technical goals for designing a mobile feedback system are portability, simplicity and efficiency. The system must be able to be used on several different client machines, the server implementation must not be tied to specific hardware, and setting up the system should not require specialized training since it cannot be assumed that all the conference lectures or moderators can be trained in its operation. The main functionalities required are:

- Ability for the audience to give feedback to the speaker using mobile devices.
- Ability for the audience to initiate communication in the form of text-based messages without requiring the speaker to specifically enable the function.
- Ability to refer to a specific portion of the presentation in the feedback.
- Ability for the presenter to immediately review and moderate the feedback.

- Rendering on a public screen to present the feedback to the entire audience.
- Ability to store the feedback for later review and to export feedback summaries.
- Both the server and client software must be cross-platform.

Based on the functionality requirements, a web-based system with multiple interfaces was designed. While it provides multiple interfaces, all the software components run on a single server machine. The server provides connectivity for mobile clients through a wireless connection. The server machine itself can display one publicly visible screen on a connected projector, or alternatively, it can be connected to a separate presentation machine.

In the context of conference presentations, different user groups use the system, each with their own objectives and requirements. The three primary user groups are the conference audience, the administrator, and the presenter. It is possible for one person to act both as the administrator and the presenter if necessary.

Figure 1 shows the user groups and presents use case scenarios possible in the system. It shows how the members of the audience mostly input new feedback while options for the presenters and the systems administrator are more extensive: They can control the stored summaries of feedback, start new conferences and pause the system if necessary.

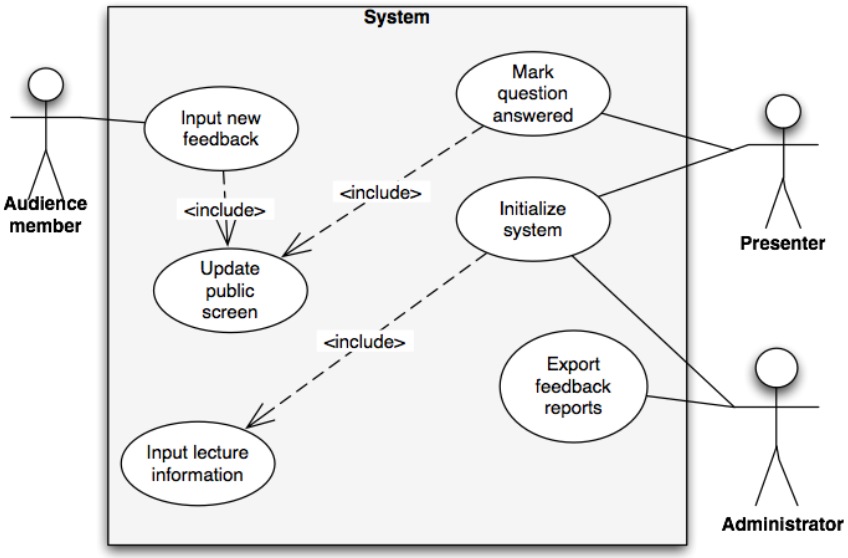

Figure 1. Use case scenarios

The differing interests of the user groups are supported in the design of separate interfaces for the groups: The audience uses the mobile interface to input new feedback and may follow the presenter’s public screen for others’ comments. The presenter directly controls the public screen with a wireless mouse that is directly connected to the feedback server or the presentation machine. The administrator uses a private web interface to control the conference program and the currently displayed lecture. For example, the administrator can set up the system to receive feedback for the next presentation after a previous one has been concluded and a new one is starting.

3.1 User interfaces

Three interfaces are available, one for each user group: A mobile web interface for audience members through which conference feedback can be given; a presenter's public screen which shows the participants' comments in real-time; and a private administration view. The public screen view has been designed to be local to the projector, usually displayed only on one machine, and can be viewed by everyone present but controlled only by the presenter. The administrator has a password-protected private view for adjusting the settings of the program. All three are accessed using a web browser.

A screenshot of the presenter view in use in a conference is presented in Figure 2. The view is divided into two parts: the header and the content. The header in the upper part of the screen view presents information about the ongoing presentation, the public web address for the feedback form, and some information about the organizer. The content of the page lists the last received comments in the order in which they were received.

The presenter can manage and moderate the content using a wireless mouse, which controls a cursor on the screen. Clicking on a comment marks it as having been answered by striking through and fading the text. In order to avoid flooding the display, very long comments are truncated by default, but can be expanded to their full size using the expand-button. For cases where a message needs to be completely removed there is a delete-button. In order to guarantee the visibility of the comments, the font needs to be large and clear enough. If there are too many comments to show them all at the same time, a scroll bar is displayed on the right side of the screen.

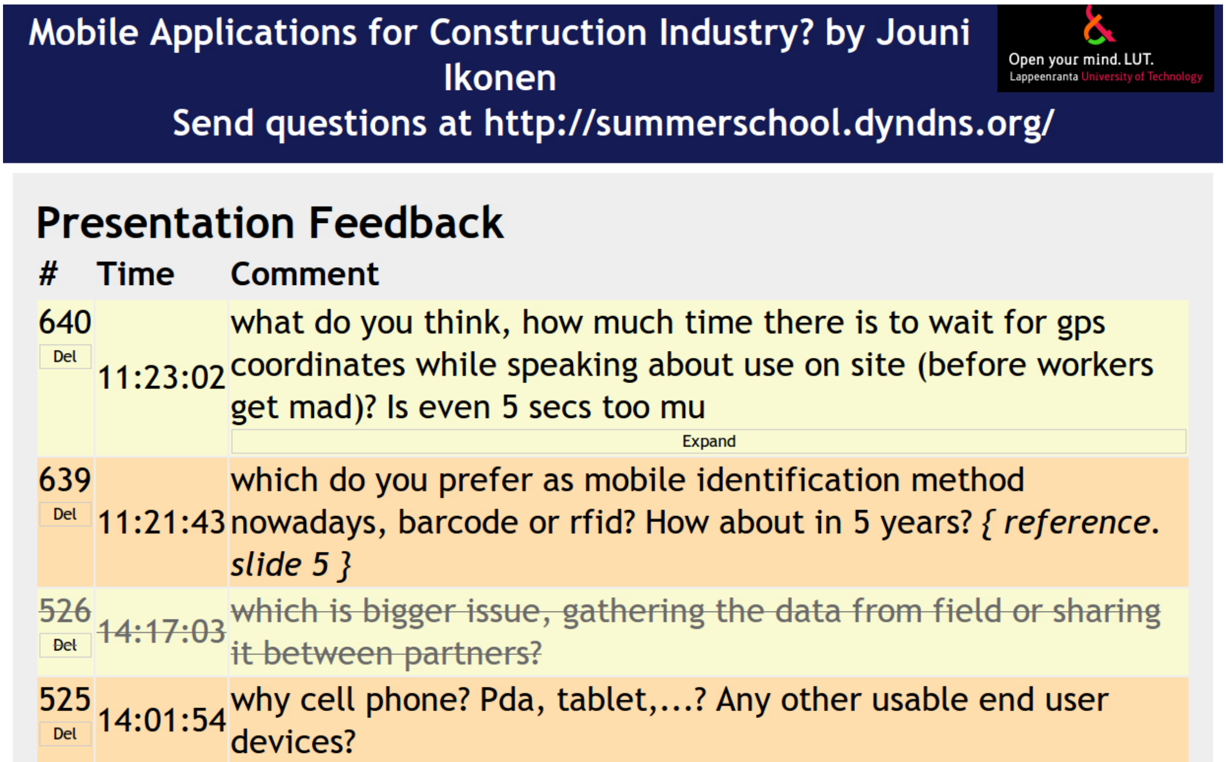

Figure 2. Presenter view

The other main interface in the system, the mobile feedback interface, is considerably simpler than the public screen view. The layout is presented in Figure 3. The mobile feedback interface consists of information about the ongoing presentation, a feedback area for typing the text, a reference that can be used to relate the feedback to a slide or a previous comment, and a list of the most recently submitted feedback.

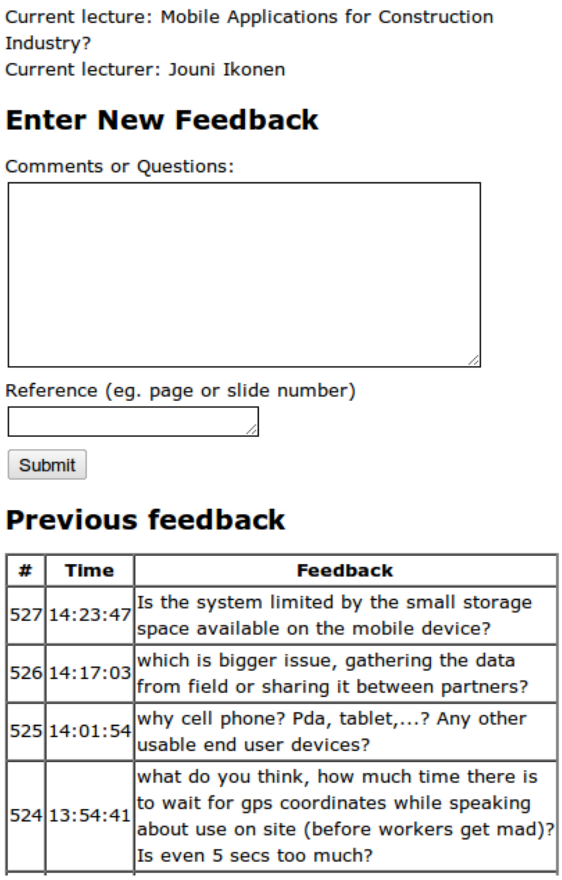

Figure 3. Mobile feedback interface

In order to maximize compatibility, all design decisions concerning the client view have been made based on low screen resolution and the lowest common feature set that can be assumed to exist on browser-equipped mobile devices. However, the interface is not tied to a specific display size and scales properly on netbooks or full-scale computers.

3.2 Application structure

The program has been implemented as PHP (PHP: Hypertext Preprocessor) web scripting language. Many web servers running on various operating systems have direct module support for PHP, thus the hardware platform is not restricted to any specific device. There are three major but loosely coupled components that are essential to the functioning of the system: The web server, the PHP application executed by client queries, and the database engine. Additionally, some of the web pages include a JavaScript component, which is loaded as a part of the page and executed on the client machine.

While the software can run on a wide variety of servers, there are certain assumptions made about the use environment and certain requirements for the available hardware. Figure 4 shows a typical configuration where all the features of a connected system are in use. The server has set up a local wireless network through which several client machines may connect to the server. Additionally the server is connected to the Internet, which enables connections through external networks. An external dynamic DNS hosting service, dynamically updated with the current address of the server, helps Internet-connected clients to discover the server. A separate presenter machine has loaded the presenter view and projects it using a connected projector.

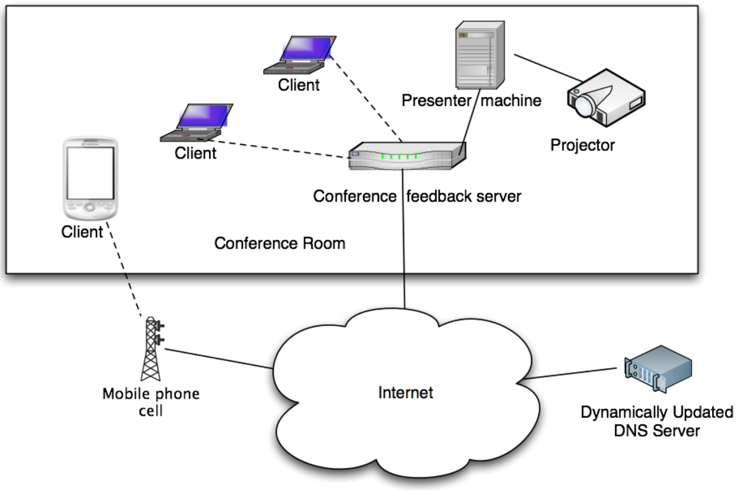

Figure 4. Typical server network environment

One server could serve several lectures simultaneously despite the software not having been specifically designed to enable it. This is made possible by the fact that PHP programs are installed by copying them into a folder in the web server. The program could be duplicated easily by copying it into several different folders, with each lecture opening a different path with the web browsers. Since all the interfaces, including the presenter view, use a browser to view it, all the features would be available. However, the software does not include management tools for that kind of operation and using single server for several lectures would lose some of the ease of configuration.

3.1.1 Server software components

The chosen database engine for the implementation is SQLite, a small c-library implementing a SQL (Structured Query Language) database engine. It was chosen for its small memory footprint and full implementation of ACID (atomicity, consistency, isolation, durability) characteristics. ACID characteristics are a set of requirements that guarantee that database transactions are processed reliably [22]. With the anticipated execution environment of the embedded devices, the resilience provided by a database that fulfills ACID requirements is important. Since the database is likely to have heavy concurrent use, initial usage tests of the system showed that the database is accessed several times per second, avoidance of access conflicts and errors due to simultaneous access is important. In addition to providing a high-performance storage engine, a full-featured database makes storing and organizing data more convenient with the functionality provided by SQL.

The only custom components are the actual PHP (PHP: Hypertext Preprocessor) program, the database schema, JavaScript running on the client browsers, and the interface markup. The web server and the database engine are open source components, chosen for their reliability. The application is compatible with practically any web server and PHP interpreter, but the SQLite database is a required component.

The server environment is multi-threaded, with a separate instance of the program stack serving each client. The different instances practically do not communicate with each other at all, ensuring a good level of concurrent processing ability. Figure 5 shows the relationships between the different server components and data interactions. The only component which all the different instances share is the database storage. The use of a shared storage for the system state ensures synchronization of all the server instances; there is no saved state between the page loads in the server instances themselves.

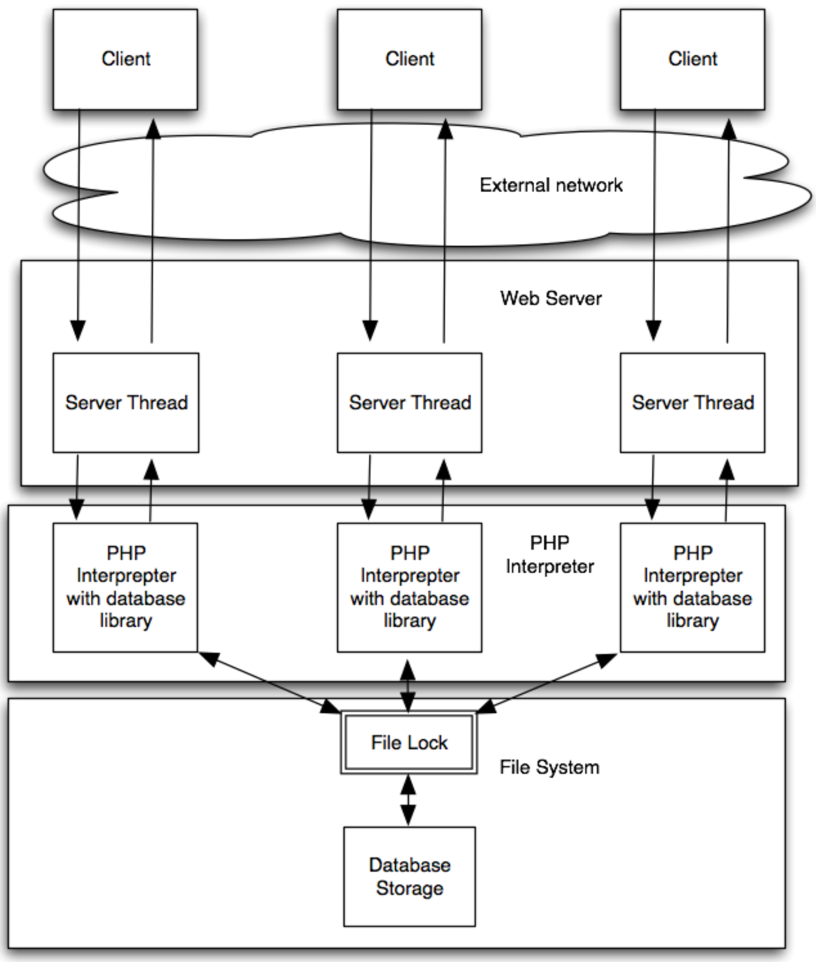

Figure 5. Server software component interactions

Third party PHP libraries and frameworks can be used to speed up development because they provide more functionality than the system libraries included with PHP. Several frameworks were evaluated, including an attempt to use the CodeIgniter framework [23] although this had to be abandoned because of performance issues on slower embedded server devices. No external libraries were utilized in the final implementation. Most of the functionality required was implemented by hand and the code was designed to perform well in the specific use cases for which the system is intended.

3.1.2 Client software components

Constant updating of the public view when a new comment is received requires that the web browser interacts with the server continuously. Since a design decision was made not to use external libraries or applications such as Java or Flash methods had to be selected to support this feature. Web clients can retrieve data from the server asynchronously using Ajax (Asynchronous JavaScript and XML) techniques. Ajax itself is not a technology but a way of leveraging existing methods to create interactive web applications [24,25]. When using Ajax communication, the web client sends a query directly to the web server. The query has an attached JavaScript listener, which waits in the background for the completion of the query and reacts when the server sends its reply. The user can keep using the browser normally while the query executes and the query does not affect the page displayed by the browser.

The greatest beneficial effect achieved with Ajax is the dynamic updating of the main public screen: Reloading the entire page is not required since the JavaScript component can issue a less complex query to the server in the background and adjust the HTML (Hypertext Markup Language) to show the latest comment on top when there is a new comment. This prevents the public view flickering from constant reloading of the page content.

The public view not only needs to be updated dynamically, it also needs to be updated rapidly to assure the impression of the comments appearing on the screen in real time. Since repetitive queries would load the server by repeatedly transferring content, a different approach was taken to updating the view: The web server hangs on the Ajax query instead of giving an immediate reply, doing rapid update polling internally from the database. This is much less resource intensive because there is no need to establish a new connection or start up a new PHP interpreter for each query.

The sequence of events that occur when the browser rendering the public screen sends an Ajax request to the server is presented in Figure 6. First the JavaScript function is executed on the client-side by a triggered timer, which initiates an Ajax query to the server. The script then becomes idle, waiting only for the server reply, while the server begins to query the local storage for as long as half a minute, checking internally for new messages several times per second. While the loop is running and no new messages have been found, no communication occurs between the browser and the server. Only once the query loop returns with new results at step three, or the loop timeouts, does the server send a reply to the client. The client also has a timeout, which is slightly longer than the server timeout, to restart the query if the server reply is lost during transmission.

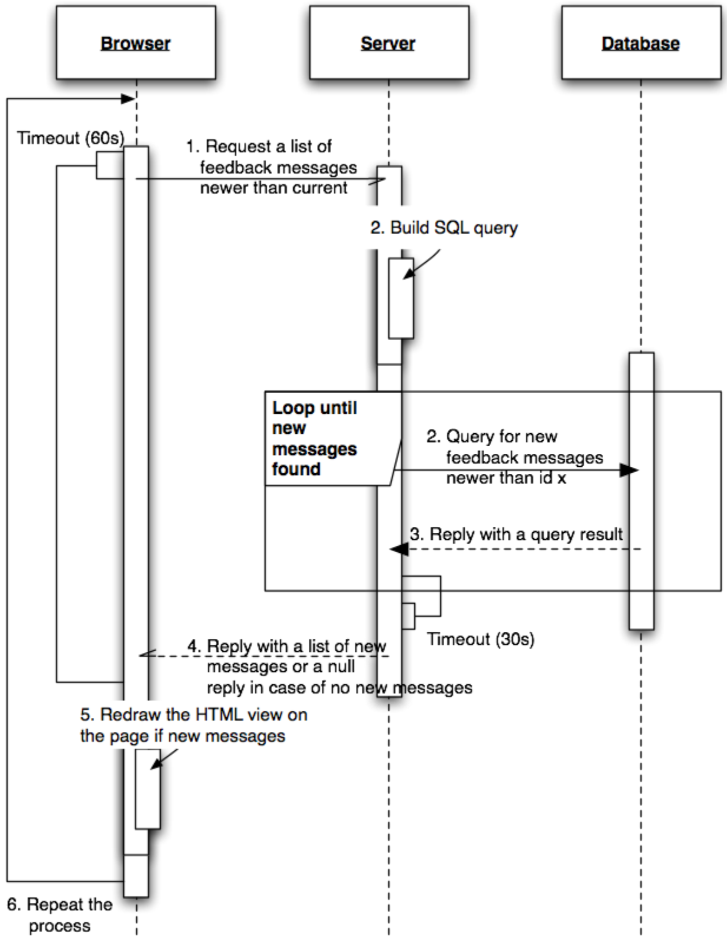

Figure 6. Asynchronous update process of the public screen

4. Case: Embedded feedback server implementation

The implementation of the system makes it possible to run the application also on an embedded device instead of a centralized web server in the Internet. The hardware platform used in the test case for the conference feedback system was a small embedded device running Linux. Although the conference feedback system can run on several different types of machine, an embedded device was chosen for its portability and affordability. The integrated WLAN (Wireless Local Area Network) access point of the device makes local client connections reliable and removes the need to set up a wireless network using additional access points in conference environments where connectivity is not necessarily provided. Connecting the device into the Internet and using third party services to link the device IP address into a static web address enables client connectivity via the Internet. The device used in this test case was GuruPlug Server [26], which is a small embedded computer with good networking capabilities and a moderately efficient ARM (Advanced RISC Machines) processor. Its small size and expandability mean the server device can be packed along with a small projector to be a quickly deployable system in a wide variety of environments.

4.1 System hardware

The main benefits of the GuruPlug Server are its compatibility with standard Linux distributions and wide networking options with Ethernet, Bluetooth and WLAN connectivity. It also has an additional advantage of being able to be connected directly to display devices. Thus, in addition to operating as a server and wireless access point, it can also display the GUI for the public view. The drawbacks are a higher price, rarity of the device and more complicated flashing procedure for installing new operating systems. As the GuruPlug can run several different distributions of Linux, setting up a web server such as Apache and a PHP module is straightforward.

The hardware used when setting up the presentation system, including the GuruPlug server, is presented in Figure 7. The setup includes the device itself and the peripherals necessary for operation, a wireless mouse for controlling the public screen, and the graphics adapter. The projector is directly connected to the server through an external USB-VGA (Universal System Bus – Video Graphics Array) adapter. In addition to being connected to the graphics adapter and the wireless mouse USB receiver, the GuruPlug is connected to the Internet through the integrated Ethernet adapter.

Figure 7. GuruPlug Server with peripherals

The default and the most basic network configuration is an isolated wireless node that has no external connectivity. If additional connectivity is provided, the device can automatically configure and share an Internet connection and upgrade a dynamic DNS (Domain Name Service) name for external access. The automatic configuration only works on typical networks and makes some initial assumptions. The preferable network environment for the Guruplug is presented in the Figure 8. If the external network has a strict firewall, e.g. blocking incoming HTTP (Hypertext Transport Protocol) requests, lacks a DHCP (Dynamic Host Configuration Protocol) server, or has other features that require custom configuration the automatic configuration fails. Providing additional connectivity through the Internet is important because some user-owned mobile client devices only have connections to external networks like the cell phone network or to WLAN networks other than the one that the server device has established.

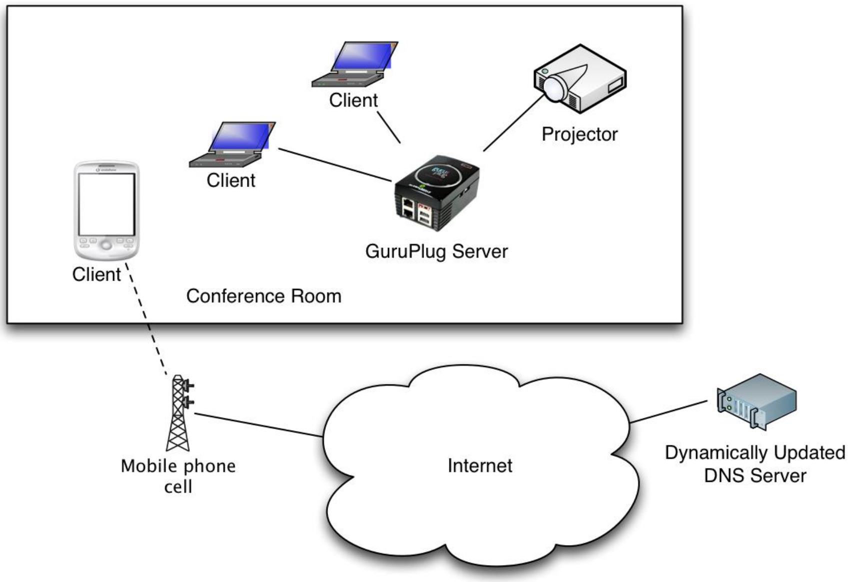

Figure 8. GuruPlug Server network configuration

4.3 System performance

The functioning and load levels of the implemented system were tested first by running the embedded device with various simulated loads and then using the device in live tests in different conferences. The performance benchmarks for the completed system were satisfactory on the test platform. The networking capabilities were monitored at a maximum of ten concurrent wireless users and the integrated access point performed well in the test setup. No heavy unrelated wireless network traffic or major wireless interference was present at the test site during the test, with the users' attention mostly on testing the system itself. If there are performance problems, they are most likely to be caused by downloads from external sites. Theoretically the available IP (Internet Protocol) addresses in the network permit a maximum of 253 users, but over thirty concurrent wireless users would saturate the bandwidth available.

The artificial performance test setup was concurrent loading of the feedback page with the ApacheBench [27] benchmarking program over a local area network running concurrently with a browser showing the public screen. This setup simulated a conference environment with multiple clients loading the audience view. Loading the page involves the following steps on the server:

1. Accepting a page request from the client
2. Passing the request to the PHP interpreter
3. Running the PHP script
4. Accessing the database and reading the ten most recent comments
5. Building the HTML code based on the database results

6. Sending the HTML code as a web page back to the client

While experimenting with various numbers of simulated clients, it was discovered that while the amount of processing time increased with the number of concurrent users, the number of requests processed per second stayed constant. In other words, the slowdown is only linear. A single page load lasted 12ms and the system could serve 4863 pages per minute. The assumed worst-case scenario for the system is 32 concurrent page loads. Since the audience is limited to a single conference hall, there should not be more simultaneous accesses as the feedback page is refreshed only when the user inputs new feedback. There was a delay of 403ms with 32 concurrent page loads and as the number of simulated users is doubled the delay grows linearly to 801ms. These are acceptable results; according to Nielsen [28] when the delay grows to over 200ms it becomes noticeable to users, but only after the delay has grown to over one second do users start to feel the waiting is unwarranted. With smaller load levels the page load wait time is under 200ms and the response feels instantaneous to users.

The system was tested in several conferences [29,30] and a random selection of users was interviewed after each test. The users replied that they felt that they had benefited from use of the system. All the lectures in which the system was used received between ten and twenty feedback messages, which can be considered a respectable number compared to the typical number of comments and questions after a presentation. A maximum of seven simultaneous mobile clients were connected to the access point during the presentations and no slowdowns in system operation were detected. In all the tests the members of the audience independently started to use the feedback system with their own mobile devices, without having explicit tutorials about the use of the system. The only instructions given were short verbal instructions about the purpose of feedback at the start of the conference and the website address displayed for the feedback site on the public screen.

5. Conclusion

The paper presented a mobile feedback system that can be used in classroom and conference situations. The system allows the audience to suggest questions during a presentation and allows the presenter to answer them at a suitable moment. Users from the audience can send their questions and comments with a PC or a mobile device. The system was not designed to replace verbal feedback but encourage a greater amount of feedback.

The project resulted in a complete system that was proven to work and can be deployed with a wide variety of platforms, including embedded server devices. The work was released as an open source project [31], which we hope will allow individuals and institutions to utilize the system for education and conferences.

While the feedback system is fully functional, there are still points for further development. While operation of the system is easy, the embedded server version does not yet adapt to new network environments as well as it might. Automatic configuration and more convenient installation to new devices would make a significant addition to ease of operation.

Live tests showed that lectures benefit from the interaction in the form of real-time feedback and that members of the audience were interested in and willing to comment on the lecture using their own mobile devices. This finding supports the initial assumption that the positive results gained from mobile device learning tools can be applied to academic conferences. A local teacher who reviewed the system raised one concern about the suitability of the system in all classrooms. The system might discourage face-to-face communication in classes that are so small where individual communication with the teacher is possible. Further testing and comparative studies would be needed to ascertain more about the impact of real-time written feedback on lectures.

References

[1] J. Rendle-Short, *The academic presentation: Situated talk in action*., USA, Ashgate Pub Co, 2006.

[2] J.T. Boyle and D.J. Nicol, Using classroom communication systems to support interaction and discussion in large class settings, *ALT-J*., Vol. 11, No. 3, 2003, pp. 43–57.

[3] P. Sullivan, Gender differences and the online classroom: Male and female college students evaluate their experiences, *Community College Journal of Research and Practice*., Vol. 25, No. 10, Dec. 2001, pp. 805–818.

[4] C. Trompler, M. Mühlhäuser, and W. Wegner, Open client lecture interaction: An approach to wireless learners-in-the-loop, *Proceedings of the 4th International Conference on New Educational Environment*, Lugano, Switzerland, 2002, pp. 43–46.

[5] H. Hämäläinen, T. Ruuska, and A. Happonen, Mobile Feedback System for Supporting Interactive Learning, *3rd International Conference on Web Information Systems*, Barcelona, Spain, 2007, pp. 446-451.

[6] M. McMahon and R. Pospisil, Laptops for a digital lifestyle: Millennial students and wireless mobile technologies, *Ascilite*, Brisbane, 2005, pp. 421–431.

[7] J.P. Mestre, W.J. Gerace, R.J. Dufresne, and W.J. Leonard, Promoting active learning in large classes using a classroom communication system, *AIP Conference Proceedings*, College Park, Maryland (USA), 1997, pp. 1019-1036.

[8] D. Frohberg, C. Göth, and G. Schwabe, Mobile Learning projects - a critical analysis of the state of the art, *Journal of Computer Assisted Learning*., Vol. 25, No. 4, Jul. 2009, pp. 307-331.

[9] R.J. Dufresne, W.J. Gerace, W.J. Leonard, J.P. Mestre, and L. Wenk, Classtalk: A classroom communication system for active learning, *Journal of Computing in Higher Education*., Vol. 7, No. 2, Spring. 1996, pp. 3–47.

[10] V. Simpson and M. Oliver, Electronic voting systems for lectures then and now: A comparison of research and practice, *Australasian Journal of Educational Technology*., Vol. 23, No. 2, Nov. 2007, p. 187.

[11] R. Valdivia and M. Nussbaum, Using multiple choice questions as a pedagogic model for face-to-face CSCL, *Computer Applications in Engineering Education*., Vol. 17, No. 1, Mar. 2009, pp. 89-99.

[12] S.W. Draper and M.I. Brown, Increasing interactivity in lectures using an electronic voting system, *Journal of Computer Assisted Learning*., Vol. 20, No. 2, 2004, pp. 81–94.

[13] C. Romero, S. Ventura, and P. de Bra, Using mobile and web-based computerized tests to evaluate university students, *Computer Applications in Engineering Education*., Vol. 17, No. 4, Dec. 2009, pp. 435-447.

[14] N. Cagiltay and S. Ozalp-Yaman, How can we get benefits of computer-based testing in engineering education?, *Computer Applications in Engineering Education*., Jul. 2010.

[15] J. Roschelle, Keynote paper: Unlocking the learning value of wireless mobile devices, *Journal of Computer Assisted Learning*., Vol. 19, No. 3, Sep. 2003, pp. 260–272.

[16] B.P. Heath, R.L. Herman, G.G. Lugo, J.H. Reeves, R.J. Vetter, and C.R. Ward, Project Numina: Enhancing student learning with handheld computers, *Computer*., Vol. 38, No. 6, Jun. 2005, pp. 46–53.
[17] R. Anderson, P. Davis, N. Linnell, C. Prince, V. Razmo, and F. Videon, Classroom presenter: Enhancing interactive education with digital ink, *Computer*., Vol. 40, No. 9, Sep. 2007, pp. 56–61.
[18] J.C. Costa, T. Ojala, and J. Korhonen, Mobile Lecture Interaction: Making Technology And Learning Click, *IADIS International Conference Mobile Learning*, Amsterdam, Netherlands, 2008, pp. 119–124.
[19] C. Locatis, Webcasting Videoconferences Over IP: A Synchronous Communication Experiment, *Journal of the American Medical Informatics Association*., Vol. 10, No. 2, Nov. 2002, pp. 150-153.
[20] A. Verma and A. Singh, Leveraging webinar for student learning, *2009 International Workshop on Technology for Education*, Bangalore, India, 2009, pp. 86-90.
[21] S.K. Wang and H.Y. Hsu, Use of the Webinar Tool (Elluminate) to Support Training: The Effects of Webinar-Learning Implementation from Student-Trainers' Perspective, *Journal of Online Interactive Learning, 7 (3)*., 2008, pp. 175–194.
[22] J. Gray and A. Reuter, *Transaction processing: concepts and techniques*., USA, Morgan Kaufmann, 1993.
[23] Ellislab Inc., CodeIgniter - Open source PHP web application framework, 2011, Available at: http://codeigniter.com/ [Accessed February 4, 2011].
[24] J.J. Garrett, Ajax: A new approach to web applications, 2005, Available at: http://www.adaptivepath.com/publications/essays/archives/000385.php [Accessed February 4, 2011].
[25] M. Mahemoff, *Ajax design patterns*., O'Reilly Media, Inc., 2006.
[26] Marvell Inc., Integrated Controller Hardware Specification, 2008, Available at: http://www.marvell.com/products/processors/embedded/kirkwood/HW_88F6281_OpenSource.pdf [Accessed February 4, 2011].
[27] Apache Foundation, ab - Apache HTTP server benchmarking tool - Apache HTTP Server, 2011, Available at: http://httpd.apache.org/docs/current/programs/ab.html [Accessed February 4, 2011].
[28] J. Nielsen, *Usability Engineering*., San Francisco, CA, Morgan Kaufmann Publishers Inc, 1995.
[29] Lappeenranta University of Technology, Summer School on Telecommunications, 2010, Available at: http://www2.it.lut.fi/comlab/ssotc/ssotc10/ [Accessed February 24, 2011].
[30] FRUCT, 8th FRUCT conference, 2010, Available at: http://www.fruct.org/node/338 [Accessed February 24, 2011].
[31] Konfeweb - Mobile Feedback System, 2010, Available at: http://sourceforge.net/p/konfeweb/home/ [Accessed February 24, 2011].