EXPRESS SURVEY WITH REAL TIME FEEDBACK

Jean-Pierre Gerval, Yann Le Ru

► To cite this version:

Jean-Pierre Gerval, Yann Le Ru. EXPRESS SURVEY WITH REAL TIME FEEDBACK. International Conference on Computers and Advanced Technology in Education, Jul 2011, Cambridge, United Kingdom. hal-02998724

HAL Id: hal-02998724

<https://hal.science/hal-02998724>

Submitted on 10 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

EXPRESS SURVEY WITH REAL TIME FEEDBACK

Jean-Pierre Gerval, Yann Le Ru

Institut Supérieur de l'Electronique et du Numérique – Brest
20 rue Cuirassé Bretagne – CS 42807 - 29228 BREST cedex 2 – FRANCE
Tel: +33 (0)2 98 03 84 00, Fax: +33 (0)2 98 03 84 10
E-mail: jean-pierre.gerval@isen.fr; yann.le-ru@isen.fr

ABSTRACT

This paper sets out a System that enables a teacher to get a real time feedback from a large group of students' understanding. A teacher can quickly set up a question and submit this question to his students. The server implements a LAMP (Linux, Apache, MySQL and PHP) architecture. We propose different types of clients that are targeting most of equipments available today: computers, mobile phone, smartphone... Thus, any student may easily find a tool to answer to any question. This System implements various types of technologies: databases, web technologies, object oriented programming...

KEY WORDS

Mobile Phone, Smartphone, Web Technologies, Object Oriented Programming, Databases, Personal Response Systems

1. INTRODUCTION

When you give a lecture to a large group of students, you cannot easily get a real time feedback of students' understanding. When you usually ask a question to them two things may occur whether nobody answers or only a few students participate and most of the time they are the same ones who did.

The problem is twofold:

- A large number of students and their timidity to answer in front of a great number of their colleagues
- Shyer students use to keep quit in large groups whereas they are not afraid to be active in small ones.

There is also many other reasons that explain why students avoid to answer questions, for example those who are afraid to give wrong answers.

The purpose of this project is to design and implement a low cost system that would enable a teacher to set up easily a question and to provide students answers in real time during the lecture.

On the one hand, most of CMS (Course Management Systems), like Moodle [1] for example, contain a survey module. It requires time and knowledge from teacher in order to create a questionnaire. It is possible to view results but the web page has to be refreshed according to new incoming answers. It requires from students to use their laptops and to login into Moodle in order to answer.

On the other hand, there are software tools, such as LimeSurvey [2] for example, which are especially dedicated to the design and the management of survey. Such tools are well suitable for large survey design but they offer too many features concerning our need that is to set up quickly a single question. Also, it does not provide results in real time even if, in the case of LimeSurvey, some interesting works have been achieved in this way [3] but limited in the field of teaching Statistics.

Personal Response Systems (PRS) such as "Clicker" [4] [5] enable interaction between students and teachers. PRS improve "the level of students' participation as well as their general attitude towards the course and their performance during tests and exams" [6]. Using Clickers sounds well suitable for small classrooms but due to its cost, close to 60€ per device, it becomes an expensive solution with very large groups of students. The main problem lies in the fact that a clicker system requires clickers!

vClicker mobile-edition [7] could be an alternative. It is web-based and runs on a Mobile Internet Device (MID): laptop, smartphone, or other portable devices with Internet browser. It enables teachers to deliver assessments to both in-class and off-site students. Unfortunately, vClicker is not Open Source and

furthermore it requires software components that are not free of charge, for example to collect and aggregate response data into Microsoft PowerPoint.

DRIM-AP [8] is a project that was achieved in order to increase lectures interactivity. Each student is using his laptop (or PDA or Tablet PC) to answer questions or to post comments through a WiFi network. However, during a lecture, it is not pleasant for a teacher to stand in front of a large amount of students hiding their faces behind their laptops screens!

Authors in [8] propose and discuss a classification of experiments and projects:

- Interactive tests: electronic votes close to clicker [9] or using SMS [10]
- Active participation : tools that target to increase teachers-students interactivity [11]
- Simulations: interactive simulations of concepts during lectures with results in real time [12]

In Japan, a cell phone-compatible comment card system [13] was developed to collect comments and questions from students. This system demonstrates the feasibility of large-scale data collect from students during university remote lecture classes.

In this paper, we focus on the development of a kind of Open Source “Clicker” which must satisfy the following requirements:

- low cost in investment and use,
- as simple as possible,
- multi-platform,
- quick and easy setup of questions,
- view results in real time,

The first part of the paper gives a brief overview of our system. In the second and the third part we describe tools that were developed respectively for students and teachers.

2. SYSTEM OVERVIEW

2.1 Introduction

An overview of the system (figure 1.) is presented hereafter. It points out the main components: server, database, management tools and client applications.

The system implements various types of clients. Those dedicated to teachers enable the management of questions and results. Those dedicated to students enable students to choose and to answer questions.

Server and database can be considered as the kernel of the System. They are using Linux, Apache, MySQL and PHP. This architecture has come to be known simply as LAMP.

Figure 1. System overview

2.2 Data Base

The Conceptual Data Model of the database is shown figure 2.

This database contains three types of objects:

1. Writers (i.e. teachers): Their login and name are automatically inserted in this table after teacher authentication. In our case authentication is implemented by means of a CAS (Central Authentication Service) Server [14].
2. Questions: This table contains the texts of questions and suggested answers that have been defined by a writer. An attribute gives the current status of the question: on-line or off-line; that is to say visible or not by students.
3. Answers: In this table are stored answers from students. A field of this table contains the IP address of the client. This is a simple solution to avoid that the same student answers multiple times the same question.

Figure 2. Database Conceptual Data Model

Two relations are linking Questions to Writers and Answers. They are useful to retrieve pertinent data from the database.

3. ON STUDENTS' SIDE

One of main ideas of this project is to enable a large group of students to reply to questions in a short time. Some of them are use to bringing their laptops in lecture hall. However, nowadays most of them have mobile phones or smartphones.

In order to enable as much as possible students to answer questions we have implemented different types of clients:

- Java clients for mobile devices,
- Web based client,
- Java clients for desktop machines.

3.1 Java clients for mobile devices

The first type of clients has been implemented using J2ME (Java 2 Micro Edition) Polish. These clients are especially dedicated to mobile phone, personal assistant and smartphone.

J2ME Polish [15] provides tools for mobile Java and cross-platform development. Main features of J2ME Polish include:

- A flexible User Interface toolkit; for example standard J2ME applications can be designed using CSS (Cascading Style Sheets) text files;
- Tools that enable the porting of an application to different equipments and also different operating systems; for example it is possible to compile the same application to an Android or a BlackBerry device;
- Software components that enable the implementation of RMI (Remote Method Invocation) for accessing server side content and communicating with remote parties;
- Software components that make easier complex data persistence on devices;
- A large database of devices.

The next two figures (Figure 3. and Figure 4.) give example of the same interface running on two different devices. This interface enables a student:

- To choose a question among available questions,
- To select his answer,
- To send his answer to the server.

Figure 3. J2ME client interface

Figure 4. Android client interface

Data transfer between clients and server are established using HTTP (HyperText Transfer Protocol) protocol. The GET method is used server side to retrieve incoming data from the client. Responses from server to client are embedded in an HTML page that is processed by the client.

3.2 Web based client

The second type of client (Figure 5.) is web based. Web pages have been designed in order to detect the user agent, which a student is using. This way the content of the page can fit different browsers from different devices: laptop, mobile phone, smartphone...

This Web client is of interest for devices that do not implement natively Java technology. This is the case for iPhone. Of course, it is possible to develop Java application targeting iPhone but it requires third party software that are not Open Source, iSpectrum [16] for example. In addition, the deployment of an application on iPhone implies that the user must download the software from the Apple Store (usually called App Store) which is not free of charge for the developer.

Figure 5. Web client interface

3.3 Java clients for desktop machines

Two other clients have been developed as standalone Java programs:

1. The first one implements the JDBC (Java DataBase Connectivity) API (Application Programming Interface). Requests to the database are prepared on client side and sent through the network to the server.
2. The second one implements RMI (Remote Method Invocation). The client is using methods from distant objects. In that case, requests to the database are prepared on server side and results are sent to the client.

4. TEACHER TOOLS

4.1 Managing questions

A web interface (figure 6.) has been implemented in order to enable teacher to manage his questions.

Figure 6. Teacher interface

The following features are available:

	Help
	Write a new question
	Delete a question
	Clear results from a question
	Edit a question
	View results
	Open a question to students
	Close a question

The last two icons (Open and Close) also acts as indicators for teacher about the current status of a question:

- Red : The question is not visible by students,
- Green: The question is visible and students can answer this question.

A presentation is available at the following address: <http://web-brest.isen.fr/sond/redacteur/jdoe.php>

4.2 Viewing results

Three technical solutions are available for teachers in order to view answers from students:

- A web page,
- A link to this web page into a PowerPoint slide,
- A Java standalone application.

The first solution for viewing results is web based. A web page (figure 7.) is linked to the above-mentioned icon "View results". Results are read from the database and updated every three seconds.

Figure 7. Results: Web viewer

A second solution stands in using LiveWeb [17]. LiveWeb is a PowerPoint add-in, which lets the user to import web pages into a PowerPoint slide. The web page can be easily inserted and can be refreshed in real time during the slide show.

The third way to view results is a standalone Java program (figure 8.). This program can be run from teacher's computer.

Figure 8. Results: Java viewer

5. CONCLUSIONS AND FUTURE WORK

This system provides teachers with a tool that enable them to ask questions to student before or during a lecture. All students can answer together in a short time and they can use various type of equipments to process their answers. Successful experiments were achieved during lectures in Computer Science and some colleagues are ready to adopt this system for their next year lectures.

Our students were involved into these developments in the framework of practical activities. It was a way to show interactions between some of the technologies that we are teaching in Computer Science: databases, web technologies, object oriented programming... On a pedagogical point of view, most of them were convinced that portability of java applications is not only a concept but also a reality. Concerning object oriented programming, it was a good opportunity for students to practice reusability and therefore to understand what it means.

Some future development could introduce a Bluetooth connection for data transfer between Java clients and server. This way there would be no need of any internet connection. This would be greatly appreciate by students who don't have WiFi access with their mobile phone or those students who have 'pay-as-you-go' plans and so, higher per-minute call costs.

We are also thinking about implementing some of the ideas that we have set out in this paper as a new module into moodle.

REFERENCES

- [1] <http://moodle.org/>
(Retrieved on January 2011)
- [2] <http://www.limesurvey.org/>
(Retrieved on January 2011)
- [3] Ewan Crawford, Neville Davies and John Marriott, A Dynamic In-class Survey Tool with Real Time Data Retrieval for Teaching and Learning Statistics, MSOR Connections, Vol. 10 No 3, Autumn Term 2010
- [4] <http://courses.csusm.edu/resources/clickers/index.htm>
(Retrieved on April 2011)
- [5] Javed Iqbal, Interactive Teaching: Use of 'Clickers' (PRS) in the Classroom,
<http://www.educause.edu/Resources/InteractiveTeaching/UseofClicke/160050> (Retrieved on April 2011)
- [6] Teresa Wrzesniewski, Use of PRS Clickers and Computer Animation in large introductory physics classes, IASTED International Conference on Computers and Advanced Technology in Education (CATE 2008), Crete, Greece, 2008
- [7] <http://www.einstruction.com/products/student-response-systems/vclicker-mobile-edition>
(Retrieved on April 2011)
- [8] Fabrice Mercier, Bertrand David, René Chalon, Jean-Pierre Berthet, Amphithéâtres interactifs dans l'enseignement supérieur, 4^{ème} colloque international sur les Technologies de l'Information et de la Connaissance dans l'Enseignement supérieur et l'industrie (TICE 2004), Compiègne, France, 20-22 octobre, 2004
- [9] Leclercq, D. et al. 1999, Votes en amphithéâtre électronique pour animer de grands auditoires universitaires selon 6 paradigmes d'apprentissage / enseignement, Actes du 16^{ième} colloque international de l'AIPU, Montréal : HEC, tome II, 567-578.
- [10] Deficall, Quizz via SMS
<http://demo.deficall.be/qcm.htm>.
(Retrieved on April 2011)
- [11] Truong T. M., Griswold W. G., Ratto M.; Star, S. L., The ActiveClass Project: Experiments in encouraging classroom participation, Technical Report CS2002-0715, UC San Diego, Department of CSE, 2002

[12] Oliver K. M., Eckel C. C., Ball S. B., Wireless interactive teaching simulations, Syllabus Conference, Santa Clara, CA, 2002

[13] Hitoshi Miyata, Improvement of classroom communication in large-scale remote lecture classes utilizing a cell phone – compatible comment card system, IASTED International Conference on Computers and Advanced Technology in Education (CATE 2005), Oranjestad, Aruba, 2005

[14] <http://www.jasig.org/cas>
(Retrieved on January 2011)

[15] <http://www.enough.de/products/j2me-polish/>
(Retrieved on April 2011)

[16] <http://www.flexycore.com/ispectrum-overview.html>
(Retrieved on April 2011)

[17] <http://skp.mvps.org/liveweb.htm>
(Retrieved on January 2011)