Questions and Answers for Designing Useful WebLabs

J. García-Zubia, D. López-de-Ipiña, P. Orduña, U. Hernández, and I. Trueba
Department of Computer Architecture, Department of Software Engineering, University of Deusto, Faculty of Engineering, Avd. Universidades 24, 48007, Bilbao (Spain). zubia@eside.deusto.es

***Abstract*—The Faculty of Engineering of the University of Deusto has had available a WebLab oriented to Microelectronics since 2001. Currently, the field of WebLab design is very active, and several other universities are adopting them as a sign of quality and distinction for Microelectronics teaching. The first part of this paper shows several requirements that a WebLab should meet and the next points discuss three remarkable aspects of WebLab-DEUSTO: a) its distinctive software-hardware architecture b) its use with a mobile telephone and c) the evaluation of its use by yhe students and the academic results they obtained.**

***Index Terms*— WebLab, Remote Laboratories, AJAX, Web 2.0**

I. Requirements for Proper WebLabs

Currently, the European university framework is changing as a result of the Bologna Declaration. Thus, the universities are trying to decentralize their activities making the classrooms and laboratories available anywhere and at anytime. Accordingly, WebLabs or Remote Labs are growing in importance and many faculties and laboratories are developing this technology to enhance their educational offer.

WebLabs may be classified in three areas according to the type of control established:

A. *Remote instrumentation*

WebLab consists of one or more experiments where users can only activate their inputs (virtual switches, signal generators,…) and see their virtual or real outputs through a webcam (LEDs, signals in a oscilloscope,…) One example of this WebLab is the Remote Access Laboratory [1] of University of Limerick and the Remote Lab of Blekinge Institute of Technology [7] (see Fig. 1), where the user can test a PLL changing the VCO control voltage, for example, and then measure PLL's output signal connected to a oscilloscope focused by a webcam.

B. *Remote parameter control*

The main difference between this WebLab type and the previous one is that here the user is able to change control parameters in order to modify the logic of the system. PID control is the most significant example of this type of WebLab, where normally the user cannot reconfigure the regulator's structure. One example of this WebLab is the Automatic Control Telelab of University of Siena [2] [3] (see Fig. 2). In this WebLab, the user can manipulate some parameters (position control, speed control, level control, flow control …) that influence the control logic of a model and the results can then be watched using a webcam.

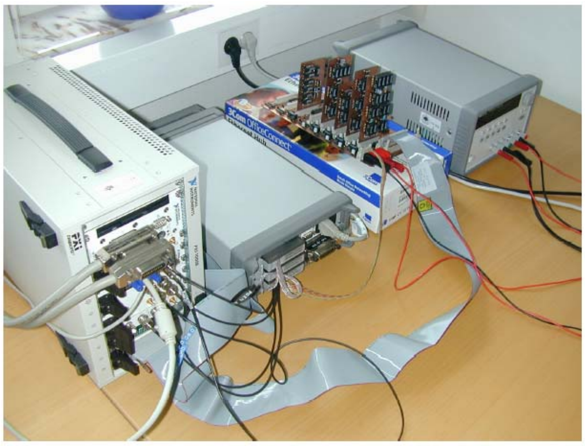


Figure 1. Remote Instrumentaion of Blekinge Institute of Technology

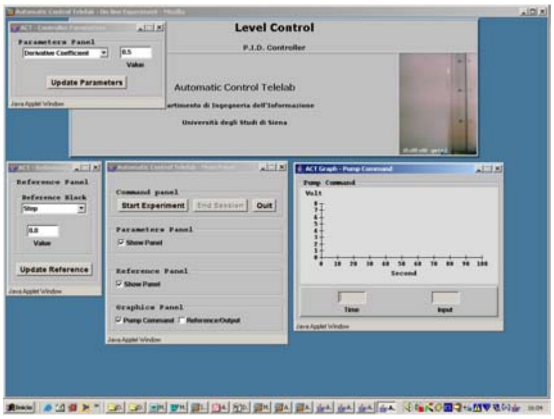


Figure 2. Remote Parameter Control of University of Siena

C. *Remote control logic*

In this case, the user can change both the logic and the system control parameters. A simple example would be a classic didactic model (LEDs, 7-segments,…) controlled by a CPLD or FPGA that has been loaded with the student program. The same could be said for a microcontroller, a DSP, a PLD or a PC-controlled system by a LabView program. Here, the risk is the destruction of the system due to a programming error because the student has the whole control over all the variables of experiment. Good examples of this type of WebLab are WebLab-DEUSTO [4] [5] (see Fig. 3), the Shell & Tube Heat Exchanger Experiment involved in iLabs of MIT [6] or the Remote Laboratory Project at the Blekinge Institute of Technology (Sweden) [7].

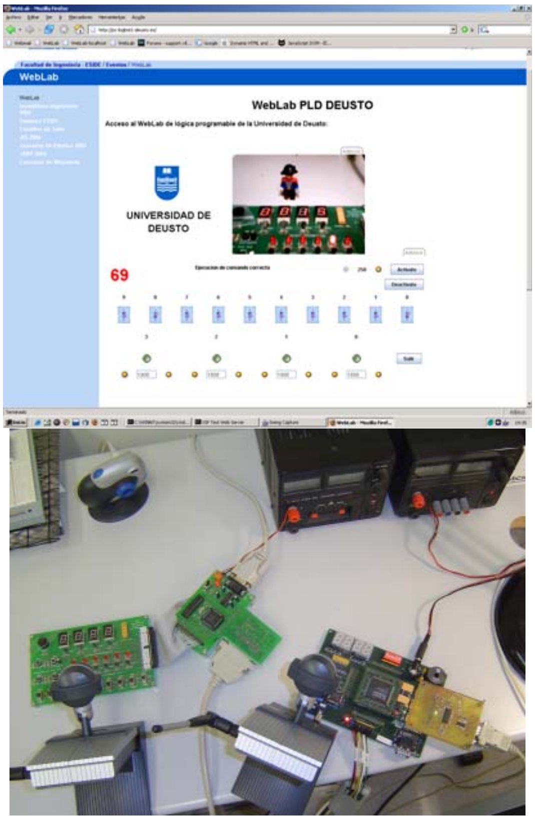

WebLab
WebLab PLD DEUSTO
Acceso al WebLab de lógica programable de la Universidad de Deusto:
UNIVERSIDAD DE DEUSTO
69
Activado
Desactivado

Figure 3. RemoteControl Logic of University of Deusto

The most powerful WebLab but not necessary the most complex, is the last one, because it includes all the previous ones. The real complexity of a Weblab depends more on didactic and service quality than on its type. The first question is very important, and many times is forgotten by some WebLab developers. However, some others authors, like Soysal [8], Ponta [9], Barron [10] and García-Zubía [5], remark its importance. The quality of service and the complexity of a WebLab depends on the following questions:

1) Is it didactic?

Is it inside of an educational platform? Does it satisfy objectives of the subject? Does it make student's work easier? Does the student feel lack of control of the WebLab? Does the student only watch the WebLab or is he involved in it? Are reports and folders generated for student and teacher? Is the quality of the algorithms controlled in order to avoid bad use and the destruction of equipments? Is it easy and visual the control of inputs, the loading of programs, watching outputs…? Are there manuals, help…available?

2) Is it universal?

Is it operative on a 24x7 basis? Is it only accessed by teachers and some guests? Is it accessed by students of other universities? Is it available in several languages? Has the server other experiments, models, robots, etc. connected? Is it possible to be simultaneously accessed? Is it allowed remote experimentation, control and reconfiguration?

3) Is it professional?

Is the WebLab designed using advanced techniques included in web 2.0? Is it integrated into university IT? Are management and administration of the WebLab (passwords, e-mail, login…) automatic? Is the time management optimum? Does the server manage the users waiting to use the WebLab? How often is the server down? Would the IT services consider that the WebLab is according with the security policies? Do the WebLab maintainers keep quality of service polls regularly?

4) Is it technologically advanced?

Does the WebLab support mobile device access? How is the communication with the controlled devices (CPLD, oscilloscopes): RS232, USB, Ethernet…? Does the WebLab depend on previously installed software on the client side: jvm, Internet Exporer, Macromedia Flash…? Is it multiplatform: Linux, Windows…? Is the quality of capturated images good? Does the WebLab rely on proprietary software: LabView ? Has the WebLab a user friendly GUI?

This list only tries to show different features of WebLabs. Each WebLab developer should try to answer those questions and make new ones. The next section shows the way in which WebLab-Deusto replies to some of these questions at the University of Deusto (Spain).

II. Software-Hardware Architecture Evolution

A correct software design is paramount in the final quality of a WebLab. The research group has been working with WebLabs for five years now. During that period four different software/hardware architectures has been designed for WebLab-Deusto

The software architecture of WebLab-Deusto has undergone through the following four iterations:

1) Socket and Applet-based Proprietary solution [11].

2) Web-based Solution [5].

3) AJAX-based Web Solution

4) Microserver-based AJAX Web Solution.

A. Socket and Applet-based Proprietary solution

The first iteration of the software architecture devised for WebLab-Deusto was a proprietary standalone client implemented in C communicating with the WebLab server through a BSD socket (see Fig. 4). This prototype was used only by lecturers and some guest students.

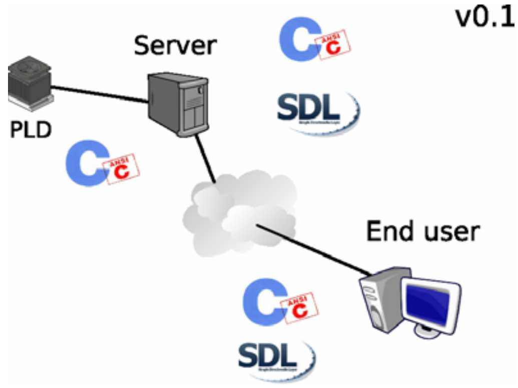


Figure 4. 1st Iteration Software Architecture

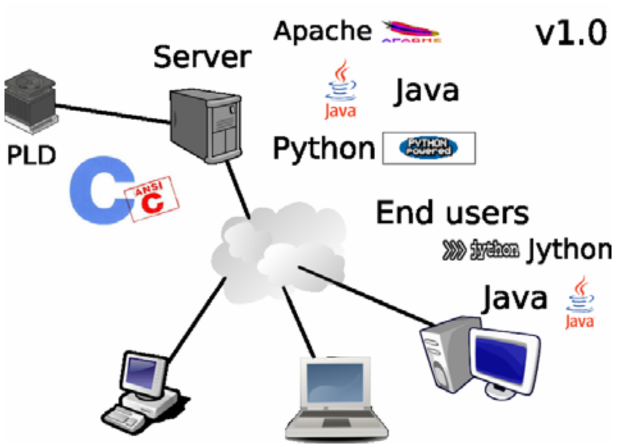


Figure 5. 2nd Iteration Software Architecture

B. *Web-based Solution*

In the second iteration, the server-side was composed of two elements: a) a Python server which communicated through the serial port with the PIC that controls a PLD and b) a webcam server broadcasting the images captured (see Fig. 5). With this iteration, students of the “Programmable Logic” subject had access to the system from an Internet browser outside the University.

The main drawbacks of this iteration were related to security. A security alert was raised every time the user downloaded the controlling applet since it required access to his PC’s file system in order to upload a file with the new programming logic. Moreover, WebLab-Deusto had to keep opened two ports in the server’s firewall: one for the webcam server and another for the controlling server. This implied an unnecessary hassle for the firewall maintenance.

C. *AJAX Web-based Solution*

This is the WebLab-Deusto currently deployed iteration (see Fig. 6). A single client application shown in the user’s browser communicates with the server through HTTP. WebLab-Deusto is a web-based firewall-safe system programmed with AJAX (Asynchronous JavaScript and XML [12]). The main benefit of AJAX is that it works on any web browser, without any plug-in installation required (thin client). The client application is now a pure HTML/JavaScript solution which follows the AJAX web interaction model. This technology is being applied successfully to sophisticated web applications such a Gmail, Google Maps or Flickr, which have been termed as belonging to next web generation, i.e. Web 2.0. The server side is composed of the elements of the previous version plus a new ASP.NET application, based on Mono, offering a SOAP Web Service interface to client applications.

In this AJAX-based web solution, the WebLab server supports both Microsoft Windows and GNU/Linux, and it does not rely on Java anymore. WebLab-Deusto also runs the client application under Nokia mobile devices running Symbian OS and equipped with an Opera browser.

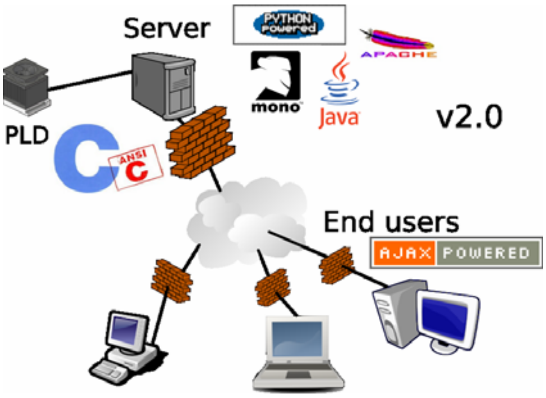


Figure 6. 3rd Iteration Software Architecture

D. *Microserver-based AJAX-based Web Solution*

Now WebLab is progressing towards the architecture shown in Fig. 7. This solution will be web-based, firewall-safe and more scalable (will provide several IP-accessible programmable devices). Many groups of users from any client platform will be able to access simultaneously any of the several networked programmable devices.

This fourth generation is again a multi-platform solution, which supports both Windows and GNU/Linux on the server-side. Moreover, there is only one programming language used: Python, maintaining the security features of the previous iterations. However, the most outstanding contribution of this iteration is the incorporation of microservers. A microserver adds to the WebLab hardware an IP address and network-based programmability. All the communication between the server and the WebLab board, previously undertaken by means of RS232 and the PIC, is now undertaken through the Internet.

The adoption of microservers opens many possibilities: use of XML (the de-facto language for data exchange), creation of autonomous WebLabs without the need of a centralized server, makes feasible the creation of intranet hardware networks and so on. Besides, the microservers are versatile, powerful and low cost (around € 100) hardware. Anyhow, the main drawback of adopting microservers is based on being a recent technology where very basic, non-sophisticated services have been deployed so far. Furthermore, it produces a dramatic change to the traditional client/server-based WebLab architecture, which can now move into a more decentralized P2P architecture. Thus, many of the functional blocks currently allocated to the server can be moved to the microservers.

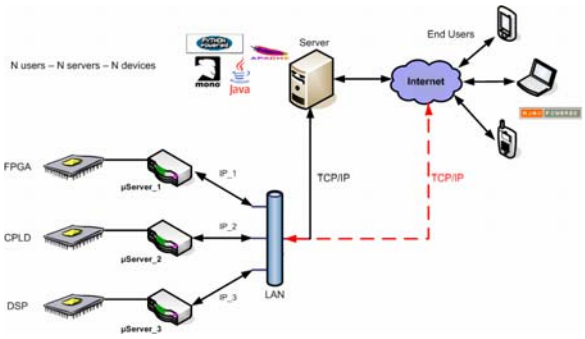


Figure 7. 4th Iteration Software Architecture

II. Characteristics of the WebLab-Deusto

Once described the technological evolution of the WebLab-Deusto, its characteristics will be described as seen in csection 1.

A. Didactics

At the moment, the WebLab-Deusto is not integrated in any educational platform such as Moodle, because the WebLab-Deusto uses a popular wiki system called mediawiki for the administration of the web site, and the WebLab-Deusto has its own manager for the system data. The most important system data is the passwords and the time management. The WebLab-Deusto does not support reservations, because every connection will last for an established time, which has previously been calculated to be enough for the designed practice. A user queue is activated when more than one student tries to use the WebLab, so the students in the queue will wait until the user who is using the WebLab finishes.

An aspect which has not been developed in the WebLab-Deusto is to control the suitability of the file uploaded by students. In the case of a programmable device, the file could set an input value to an output, or it could try to use the wrong FPGA model, causing the destruction of the programmable device. Controlling this would equip the WebLab with certain semantic abilities.

Section 5 will treat educational effectiveness in depth.

B. Universal

The WebLab-Deusto is available 7 days per week, 24 hours per day, both for students as for guests, in english, spanish and euskara (regional language).

At the moment, the WebLab-Deusto is centered in the third type of WebLab following the classification of section 1. This is, the WebLab-Deusto is oriented to remotely offer equipments that need the “logic” to operate, being this logic what the students have to design, in different devices (ie. CPLD, FPGA, microcontrollers, robotic, DSP, etc.) WebLab-Deusto is not oriented to remote experiments (type 1) with oscilloscopes, function generators, etc.

The WebLab-Deusto currently does not support the simultaneous access to the practice by several students, because several students changing the logic of the device concurrently makes no sense. Anyway, this approach does not allow the collaborative work. In this educational approach, several students would design a VHDL program through a wiki and a chat or skype, and one of them would upload the logic while the rest of them look at how the practice progresses to check if the program is working correctly or what changes should be added. The WebLab-Deusto must allow simultaneous access to the practice in the future.

C. Professional

The most important challenge for the WebLabs consists in being accepted as a fundamental part of the structure of the university, at the same level of Internet, the intranet, etc. This is, the WebLabs have to leave the laboratories in order to be included in the infrastructure of the university. To reach this point, the WebLabs must fulfill some requirements so as to achieve a more professional architecture.

The WebLab-Deusto is implemented in AJAX, following the Web 2.0 guidelines. It is cross-platform (it runs at least under Microsoft Windows, GNU/Linux and Mac OS X), and the client only needs a web browser (such as Mozilla Firefox, Microsoft Internet Explorer, Opera browser...), without any need of a plugin or any installation by the user. The WebLab-Deusto uses SOAP through the HTTP protocol, so neither the client or the server need to open more ports than the HTTP and HTTPs ones, which are usually open anyway. And, as all the sensitive information goes encrypted through HTTPs, the security problems derived from the connection are minimized.

The most important sign of professionalism of the WebLab-Deusto from the preceding paragraphs is granting security from the point of view of the IT services of the university. In not few cases, the WebLab design demands several ports to be opened, which obstructs the security policy of the university. It is not weird to hear that for showing a demo in a different university, the WebLab designer ask for all ports to be opened, which is a non welcome policy by IT administrators.

In the same line, there are problems with plug-ins. Many WebLabs ask users to install a plug-in in order to be able to run use the WebLab. This sometimes is fine, but in many situations it can be a drawback. Firstly, some of these plug-ins are not available on every platform, so if the clients find themselves in those situations, they simply can not use the WebLab. Secondly, the plug-in might be available for the platform of the client, but he could be under an account with few privileges, being impossible to install the plug-in. This situation is pretty common: any university which, for security reasons, doesn't allow students (and even professors) to have an administrator account in the computer laboratories would perfectly fit in it. This restriction might affect both particular and commercial plug-in such as Java Runtime Environment. A third restriction would be the compatibility between plug-ins. The client might have a Java Runtime Environment or the Macromedia Flash plug-in already installed, but if the WebLab relies on a higher version of these programs, it will just not work. As a conclusion, the thinner the client is, the further it will reach.

D. Advanced technologies

More and more, client/server applications are migrating to mobile devices such as PDAs, cellular phones, etc. The WebLab-Deusto design allows the remote use using a cellular phone being notable the fact that the design did not need to be revised to achieve it. The same server that serves a PC serves in the same way the user behind a cellular phone. This situation derives from the correct use of the last technologies.

III. Accessing the WebLab trough a Cellular phone

Mobility 2.0 is trying to bring Web 2.0-enabled applications to the mobile domain. In a nutshell, it attempts to translate the benefits experienced by users of Web 2.0 applications on the desktop to mobile devices.

There are already some good examples of Mobility 2.0-enabled applications. For example, Google Local for Mobile (http://www.google.com/gmm/index.html) enables a user to access GoogleMaps from a Java enabled mobile running a J2ME application; Yahoo! Go Mobile (http://go.connect.yahoo.com/go/ mobile) is a J2ME client which enables a user access to Yahoo! Services such as Contacts, Email, Photos or Messenger from mobile devices; or even some Mobile Blog clients (Mobile Blog, Blogger) which populate blogs from mobile devices. All those applications are regarded as *mobile mash-ups*, i.e. they are web applications adapted to mobile devices combining content from several sources into an integrated experience. In order to develop such mobile mash-ups two main models have been followed:

A. *Browsing apps*

Web apps which take into account limitations unique to mobility (e.g. small device or network bandwidth). The XHTML clients are capable of hardly any processing.

B. *Smart client apps*

Downloaded and installed in the device. Clients are capable of some processing, storage and intermittent communication. Some example enabling technologies are J2ME, Compact.NET, Python for Series 60, BREW uiOne or Flash Lite.

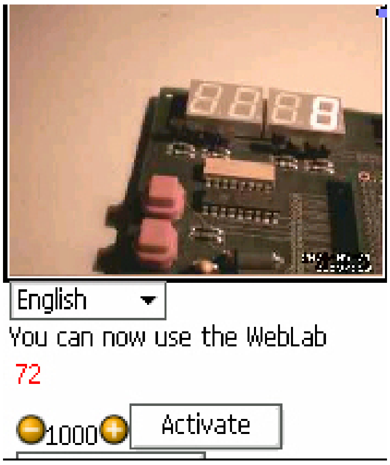
AJAX is a very important facet of Web 2.0. It avoids start-stop cycles emitting asynchronous calls to the server, so that the user does not wait. It solves two problems: a) superior UI experience and b) standardized form of data retrieval.Unfortunately, it does not have much presence on mobile devices. However, this situation is changing since many last generation mobiles come equipped with Opera Browser or Internet Explorer for Windows Mobile 5 which both support AJAX.

Following this emerging trend on the development of mobile applications WebLab-Deusto has been designed to be accessible from mobile devices, i.e. to turn into a Mobility 2.0 application.

As described in section 2, the client of the WebLab-Deusto is just a common web front-end running on a browser. It does not rely on any proprietary plug-in for the web browser such as Java Applets or Macromedia Flash. Thus, any Web Browser which implements the commonly used web standards required in AJAX is a potential client of the WebLab-Deusto. Due to this fact, it is easy to find clients running on different platforms, including Microsoft Windows, Mac OS, GNU/Linux or even mobile devices (Symbian, Windows Mobile).

The Opera web browser is a proprietary AJAX-compatible software available under many mobile platforms, like Nokia S60, S80, S90, SmartPhones with Windows Mobile, and so on. Any mobile device running the Opera Browser or the latest edition of Internet Explorer for Windows Mobile (also AJAX-compatible) can access the WebLab-Deusto (see Fig. 8), without changing anything in the architecture of the WebLab-Deusto. Currently, there are many ongoing projects aiming to develop both proprietary and open source AJAX-compatible web browsers for these devices. Therefore, it is feasible to assume that in the near future all mobile devices will be equipped with such browsers.

The other approach to access a WebLab from a cellular phone is programming a specific proprietary client for the mobile device. In order to avoid losing portability between different mobile devices, there are cross-platform development platforms available, mainly J2ME and Compact .NET. If WebLabs use a standard protocol for communication as is the case of WebLab-Deusto with SOAP, the development of clients with any of these technologies is simple. In fact, following this approach WebLab developers can, at the time being, aim at a wider range of devices.



You can now use the WebLab

Figure 8: WebLab-FPGA from Opera Web Browser on a Nokia 6630

The main drawback of the proprietary client approach is that it obviously requires the development and maintenance of a new client for any mobile platform supported. Moreover, every new feature in the WebLab should be ported to both the Web client and all the platform proprietary clients. With the AJAX approach followed in WebLab-Deusto, the user would automatically access the last version of the WebLab client every time she enters the WebLab's web site.

IV. Academics

Currently, the AJAX-based WebLab-DEUSTO (http://weblab.deusto.es, http://weblab-pld.deusto.es, http://weblab-fpga.deusto.es) is used for student assignments which require access to a CPLD or a FPGA (Xilinx CPLD and FPGA XC2S144). In particular, the assignments for the subjects “Programmable Logic” and “Electronics Design” of the third and fifth year of Automation and Electronics Engineering, respectively, are carried out with the help of WebLab-DEUSTO.

Table 1 summarizes the results of a questionnaire given to the students. Grading system goes from 1 to 5. Question 9 is of special interest. The student indicates that even if he is far away from the prototype he does not feel that he has lost control of it, in other words, the student feels that the assignment is “his”.

V. Conclusions and Future Work

This work has provided three main contributions. Firstly, it is very important to pay extra attention to the software-side of a WebLab design, even more than to the hardware-side, since many problems of deployed WebLabs come from poor software-side designs (accessibility, security, and so on). Secondly, the use of microservers on the hardware-side will revolutionise and encourage the usage and design of WebLabs. Thirdly, academically it is obvious that the use of a WebLab improves the subject teaching and the opinion that students have about the labs, the subjects and the lecturers. Anyhow, it is always important to control the quality of new developments in a WebLab, checking the students’ opinion. Currently, the research group is working in three aspects: a) extending the use of WebLab-DEUSTO to microcontrollers and DSP, b) redesigning WebLab-

DEUSTO by adopting microservers and c) documenting the academic performance of WebLab-DEUSTO.

TABLE 1. RESULTS OF A QUESTIONNAIRE PPROPOSED TO THE STUDENTS

Questions	Average (1)	Average (2)	Average (3)
Number of acceses to the WebLab	1.706	495	632
1. Has WebLab helped you with the subject?	4.6	4.1	3.8
2. Did you feel that you were in a better position by having been in the WebLab group?	4.7	3.9	3.9
3. Do you think it is a good idea if this WebLab experiment is extended to all the students?	4.7	4.6	4.2
4. Is it easy to use?	4.4	4.4	3.9
5. What is the quality of the WebCam like?	3.2	2.4	2.7
6. Did you feel at ease managing the inputs?	3.7	3.1	3.0
7. What do you think about the time assigned to each connection?	3.7	2.7	3.1
8. What do you think about the inputs/outputs implemented?	3.8	3.2	3.4
9. Being far from the prototype, have you felt you were in control of it?	4.1	3.7	3.6
10. Would you like to use WebLab in other subjects?	4.3	4	3.9
11. What is your global satisfaction with WebLab?	4.7	3.9	3.7

(1) Results in 2004/2005 for the subject "Programmable Logic" in the third course.

(2) Results in 2005/2006 for the subject "Electronics Design" in the fifth course.

(3) Results in 2005/2006 for the subject "Programmable Logic" in the third course.

REFERENCES

[1] http://www.ral.ul.ie/?main

[2] http://www.dii.unisi.it/~control/act/introduction.php

[3] Casini, M.; Prattichizzo, D. y Vicino, A. (2003) "e-Learning by Remote Laboratories: a new tool fo control education". The 6th IFAC Conference on Advances in Control Education, Finland

[4] http://weblab.deusto.es

[5] García-Zubia, J et al. "A new approach for implementing remote laboratories: a practical case". 2nd International Symposium REV 2005, Brasov (Romania), July 2005, ISBN: 3-89958-137-7

[6] Alamo, J.A., MIT Microelectronics Weblab, March 27th, 2001. http://web.mit.edu

[7] Gustavsson, I. et al. "A flexible remote electronics laboratory". 2nd International Symposium in Remote Engineering and Virtual Instrumentation, REV 2005, Brasov (Romania), July 2005

[8] Soysal, O. "Computer Integrated Experimentation in Electrical Engineering Education over Distance" Proceedings of ASEE 2000 Annual Conference, Saint Louis, MO, June 2000.

[9] Ponta, D. et al "Remote experiments in electronics: pedagogical issues" 2nd International Symposium REV 2005, Brasov (Romania), July 2005, ISBN: 3-89958-137-7

[10] Barrón, M. "Laboratorios virtuales para enseñanza por internet" en I Jornadas de Tendencias sobre eLearning, TEL 2005, Madrid (Spain), February 2005

[11] García Zubía, J. "Programmable Logic and WebLab", Proceedings of the 5th European Workshop on Microelectronics Education, Lausanne (Switzerland), April 2004, ISBN: 1-4020-2072-4, PP: 277-282

[12] http://en.wikipedia.org/wiki/AJAX

AUTHORS

J. García-Zubia is with the Faculty of Engineering of the University of Deusto, Bilbao, Spain (e-mail: zubia@eside.deusto.es).

D. López-de-Ipiña is with the Faculty of Engineering of the University of Deusto, Bilbao, Spain (e-mail: dipina@eside.deusto.es).

P. Orduña is with the Faculty of Engineering of the University of Deusto, Bilbao, Spain (e-mail: pablo@ordunya.com).

U. Hernández is with the Faculty of Engineering of the University of Deusto, Bilbao, Spain (e-mail: uhernand@eside.deusto.es).

I. Trueba is with the Faculty of Engineering of the University of Deusto, Bilbao, Spain (e-mail: itrueba@eside.deusto.es).