Experience with WebLab-Deusto

Javier García-Zubía, Diego López-de-Ipiña, Pablo Orduña, Unai Hernández-Jayo
Faculty of Engineering, University of Deusto, Apdo. 1, 48080 Bilbao (Spain)
zubia@eside.deusto.es, dipina@eside.deusto.es, pablo@ordunya.com, uhernand@eside.deusto.es

Abstract: The Faculty of Engineering of the University of Deusto has made available a WebLab oriented to Microelectronics since 2001. Currently, the field of WebLab design is very active, and several universities are adopting them as a service of quality and distinction for Microelectronics teaching. This paper describes two main aspects of our WebLab, namely WebLab-DEUSTO: a) its software-hardware architecture and b) the academic results obtained by its users.

1. INTRODUCTION

The concept of WebLab has been around since the early nineties. Its development is widespread in laboratories of analog [12] and digital [5] electronics, programmable logic [8] or process control [14]. Good examples of WebLabs can be encountered in different countries: USA [1], Spain [9] [13], Portugal [3], Italy [7], Corea [14] and so forth.

A WebLab can be studied from different points of view:

- *Didactical*: didactic goals, quality and suitability of the WebLab, didactic platform integration, etc. [7], [6], [9] [16].
- *Hardware technology*: cards, electronic prototype, data acquisition, etc. [6] [12]
- *Software technology*: client/server design, security, integration, etc. [6]
- *Software development platforms*: Web-Services [14], LabView [5] [10], C applications [2], JAVA [15], Matlab [3], etc.
- *Communication*: through RS-232 [11], TCP/IP [9], XML [4], etc.
- *Social*: international solidarity, disabled people adaptation, etc. [8]

The recent popularity of the WebLab concept, its different approaches and the abundant existing bibliography only prove the great activity on a field which is called to represent a cornerstone of worldwide engineering education.

WebLabs are traditionally designed by electronic and control engineers who naturally tend to place a major attention on the hardware side of the system. They usually follow a three step process: (1) choose a programmable device, (2) attach it to a server, accessible through the web or simply a TCP/IP socket, and (3) design a simple protocol to record programs in the remote device, send inputs and receive outputs. Unfortunately, the software side involved in the last two steps is often paid too little attention and hence a poor usage of the remotely available programmable hardware devices is achieved. Better software architectures for WebLabs should lead us to more user-friendly, cost-efficient, reliable and scalable WebLabs.

The structure of the paper is as follows. Section 2 shows the evolutions of the software architecture of our WebLab and proposes a new "ideal" architectural model for WebLabs. Section 3 stablish the current incarnation of our WebLab at the University of Deusto and its academic results. Finally, section 4 draws the conclusions and the future work.

2. SOFTWARE ARCHITECTURE EVOLUTION

The software architecture of our WebLab has gone through the following four iterations:

1. Socket and Applet-based Proprietary solution [8].
2. Web-based solution [9].
3. AJAX-based Web solution [9].
4. MicroServer-based AJAX-based Web solution.

As a result of this iterative process we have envisioned the architecture of a next-generation WebLab which will allow mainstream access to WebLabs worldwide, we have called this architectural concept "Universal WebLabs".

A. Socket and Applet-based Proprietary Solution

The first iteration of the software architecture we devised for our WebLab is shown in Fig. 1. A proprietary standalone client implemented in C communicated using the SDLnet library with the WebLab server. This server was in charge of communication through RS-232 with a PIC acting as bridge of a programmable PLD. In parallel to the command-line application remotely controlling the programmable device, an ActiveWebCam applet by PySoft was used to observe in real-time the status of the hardware being programmed. The WebLab server kept user-access and usage control. Each time only one user could be accessing the remote device. This was a prototype only used by lecturers and guests.

The main drawbacks of this solution were:

- *Interoperability issues*. Both the client and server solutions could only be run on the MicroSoft Windows platform.
- *User-friendliness issues*. The users needed to start two independent applications, the controlling standalone C-based application and the Java viewing applet. Moreover, the controlling client offered a primitive command-interface through which FTP-like commands could be used to upload new logic to the programmable device, induce inputs and read outputs.

- *Security issues*. On the server side, the firewall has to be configured to enable traffic offer two non well-known ports rather than using already opened ports such as 80 for HTTP. In addition, there was not built-in user access control. Consequently, there was fear to open the WebLab to the public, and it was only used for demonstration purposes within the University's LAN.

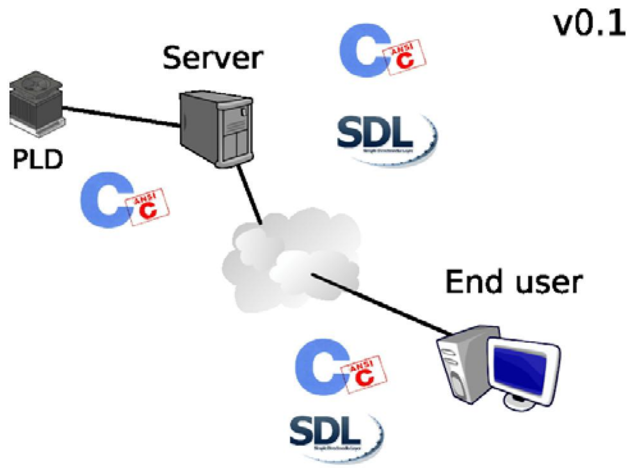

Fig. 1. 1st Iteration Software Architecture

B. Web-based Solution

The second iteration of our software architecture is shown in Fig. 2. Here, the server-side was composed of three elements: a) an Apache web server hosting a webpage with the controlling and viewing applets, b) a Python server which communicates through the serial port with the PIC that controls a PLD and c) a webcam server broadcasting the images captured. In this iteration, the client application was totally based in Java, accessible through a web browser with a pre-installed Java plug-in. The controlling applet communicated with the controlling server, whereas the viewing applet connected with the webcam server.

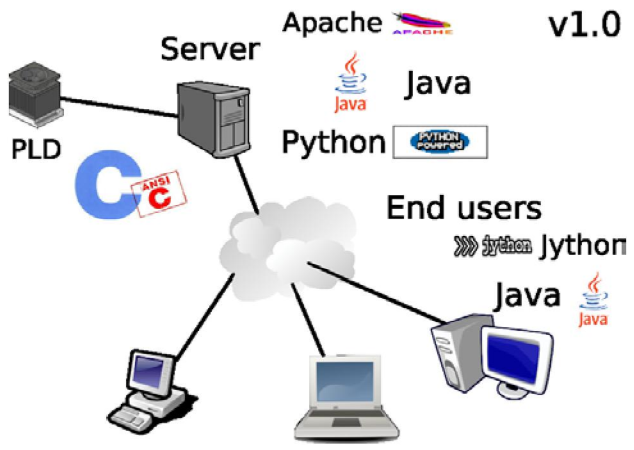

Fig. 2. 2nd Iteration Software Architecture

The WebLab server's logic was updated to keep user-access and usage control. Each time only one user could be accessing the remote device for a maximum period of time (120 secs). The only requirement imposed to students was to use a browser with a pre-installed Java plug-in.

This solution still presented some issues regarding user-friendliness and security:

- *User-friendliness issues*. We had two independent applets executing on the same webpage. The download of the applets took some time and required the user browsers to have installed the Java plug-in.
- *Security issues*. A security alert was raised every time the user downloaded the controlling applet since this required access to the file system of the user in order to upload a file with the new programming logic. Moreover, we still had to keep opened two ports in the firewall: one for the webcam server and another for the controlling server. This supposed a hassle for the firewall maintenance.

With this iteration, we finally gave access to students of the "Programmable Logic" module to access the system from an Internet browser outside the University.

C. AJAX-based Web Solution

The third iteration of our WebLab, currently in use and shown in Fig. 3, is a single client application shown in the user's browser communicates with the server through HTTP. We now have a web-based firewall-safe system programmed with AJAX (Asynchronous JavaScript and XML). The main benefit of this web development approach is that it only uses tools readily available on any web browser, i.e. XHTML, DOM and JavaScript. Therefore, no plug-in installations were required on the users' browsers.

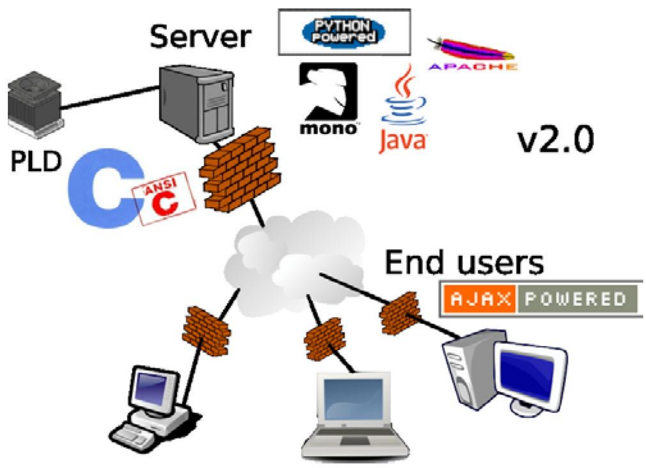

Fig. 3. 3rd Iteration Software Architecture

The server side is composed of the following elements: a) a Java server continuously capturing images from a WebCam and saving them into a directory exported by an Apache web server, b) a Python server controlling the communication with the programmable device and c) an ASP.NET application based on Mono and running on the Apache Web Server offering a web-service interface to client applications.

The client application is now a pure HTML/JavaScript solution which follows the AJAX web interaction model, i.e. rather than changing the full content of a page every time there is an interaction between the client and server, only the portion of the page affected by the interaction is modified. This technology is being applied successfully to sophisticated web applications such a Gmail, Google Maps or Flicr. The key of this technology is that the control commands, responses and images are transmitted asynchronously, without interrupting the user interaction with the system, by means of a JavaScript's XmlHttpRequest object.

The data exchanged between the AJAX client and the Mono-based server is through the standard Web Services transport protocol, namely SOAP. The Mono-based server delegates the arriving web-service method invocations to the Python server controlling the programmable device. The latest captured image is continuously being retrieved through HTTP by the AJAX-based client by accessing to a well-known URL.

The main drawbacks of this solution are:

- *Interoperability issues*. Although the client-side is multi-platform, the server software still relies on the Windows platform. Both the serial communication and storing software programs only run on Windows.
- *Server Software Maintenance issues*. Far too many technologies are used on the server side: Java, Python and ASP.NET. For maintenance purposes it would be interesting to concentrate all the functionality in a single component developed with only one programming technology.
- *Image Streaming issues*. The reception of the remotely programmable device images is still far from optimum. Each image is transmitted as a JPEG file instead of a streaming solution which would allow for a more up to date and reliable tracking of the remote device's activities.
- *Security issues*. Now only port 80 is used in the communication between the client and server side of the system. Therefore, this solution is firewall-safe.

D. MicroServer AJAX Web-based Solution

We are currently to the WebLab architecture shown in Fig. 4. This solution will be web-based, firewall-safe, more scalable (will provide several programmable devices) and support cooperative work among group members. N groups of users from any client platform will be able to access simultaneously to any of the N networked programmable devices.

In our third WebLab iteration, the communication and control of I/O was performed through RS-232 by means of a PIC microcontroller acting as a bridge between the server and the electronic prototype. Moreover, the WebCam was connected to the server by means of an USB port. Therefore, if we wanted a single server to control several prototypes and WebCams we would need several serial and USB ports together with the corresponding coordination protocol for all those devices.

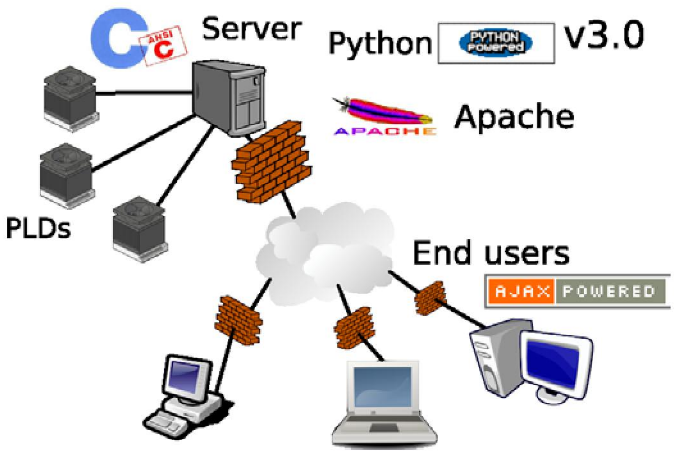

Fig. 4. 4^{th} Iteration Software Architecture

In the currently ongoing development of the fourth iteration of our WebLab we will replace the PIC microcontroller by an assortment of IP-accessible MicroServers. The adoption of microservers will turn our WebLab into a much more flexible and scalable distributed system:

- The WebLab server will no longer have to deal with the low-level RS-232 communication details. It will instead communicate through HTTP by means of data encoding standards as XML.
- The MicroServers will allow the set of programmable devices within a WebLab to be connected in a LAN. The MicroServers will connect either through an Ethernet port or will host an IEEE 802.11 chip to allow them to be wirelessly connected among themselves and the controlling WebLab server.
- The electronic prototypes attached to the MicroServers will also be capable of exchanging information among themselves. The information does not only flows between the prototype and the server, but it also can flow among prototypes with the help of the MicroServers.

With the incorporation of MicroServers each programmable device in a WebLab will be transformed into a networked node. Therefore, network administrators will now have to deal with a new type of device and ensure it is operational on a 24x7 basis.

In conclusion, this fourth iteration will provide us with a secure cross-platform (see Fig. 5), which maximises the use of the hardware resources allocated. The core idea behind our fourth iteration architecture will be to "push away" from the WebLab server all the functionality specific to a given programmable device. In our opinion, this solution will approach to the ideal software architecture for a WebLab.

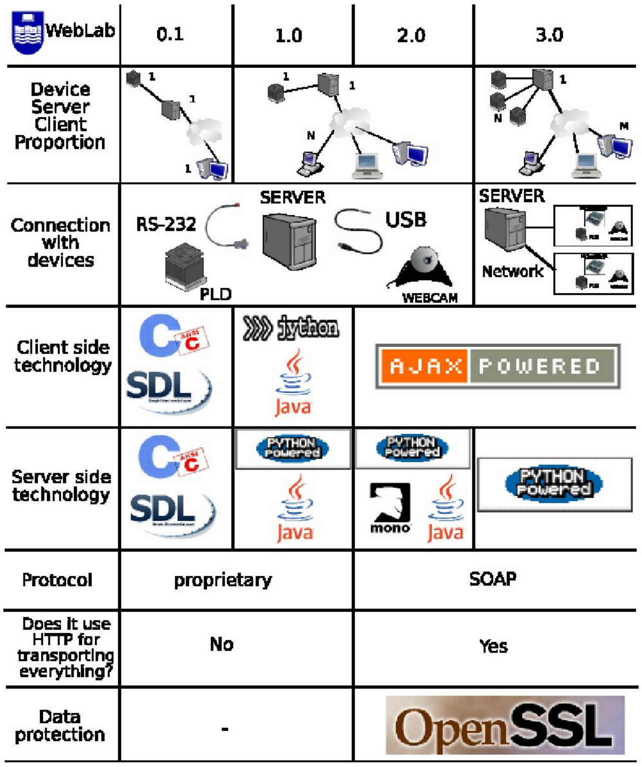

Fig 5. Evolution of WebLab-PLD.

3. ACADEMIC RESULTS

Currently, WebLab-DEUSTO (http://weblab.deusto.es) serves us to carry out assignments over CPLD y FPGA (CPLD de Xilinx XC9572 y FPGA de Xilinx XC2S144). The WebLab-DEUSTO is used in the subjects "Programmable Logic" in the third year of Automation and Electronics Engineering and in "Electronics Design" of the fifth year of the same degree. See Fig. 6, 7 and 8.

During the second semester of 2004/2005 course WebLab-PLD was in use by the students registered in the subject "Programmable Logic". This course has 90 students registered, out of those 65 chose to do their practices using a regular laboratory and 25 have done theirs using WebLab. Out of the 65 "regular" students, 16 have failed and 49 have passed, that is 75% passed and a 25% failed the practical part of the subject. Out of the 25 students using WebLab, 2 have failed and 23 have passed, that is a 92% pass and an 8% fail. These results clearly show that WebLabs have a beneficial effect on students. The WebLab only crashed twice in the semester.

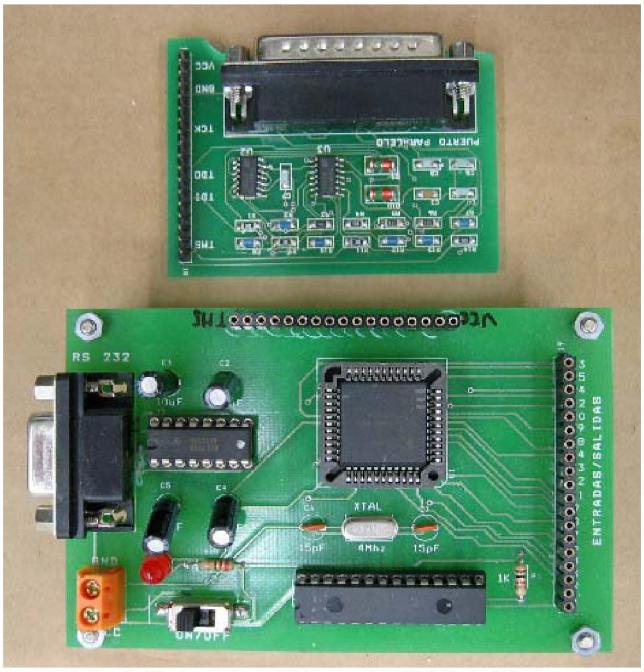

Fig 6. University of Deusto's WebLab-PLD

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

TABLE 1.
RESULTS OF A QUESTIONNAIRE PROPOSED TO THE STUDENTS

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

(1) Results in 2004/2005 for the subject "Programmable Logic" in 3rd course
(2) Results in 2005/2006 for the subject "Electronics Design" in 5th course
Table 2. Results of a questionnaire proposed to the students

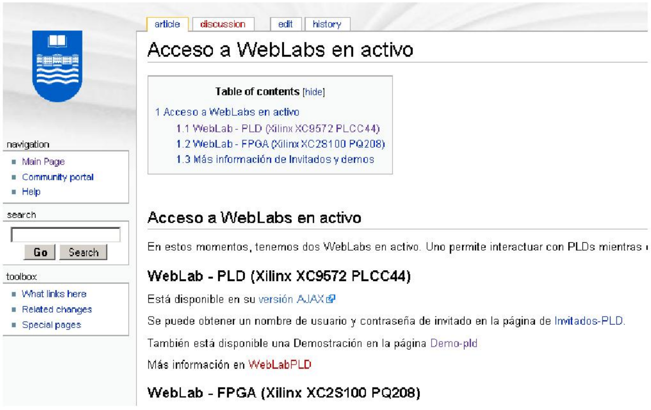

Fig 7. http://weblab.deusto.es.

Fig 8. WebLab-Deusto demo.

4. CONCLUSIONS AND FUTURE WORK

This work has provided three main contributions. Firstly, it is more 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 the 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, our research group is working in three directions: a) extend the use of WebLab-DEUSTO to microcontrollers and DSP, b) redesign WebLab-DEUSTO by adopting microservers and c) document the academic performance of WebLab-DEUSTO.

7. REFERENCES

[1] Alamo, J.A., (2001) MIT Microelectronics Weblab. http://web.mit.edu

[2] Aliane, N.; Martínez, A.; Fraile, A.; Ortiz, J. (2005) "LABNET: laboratorio remoto para control de procesos". *Actas de las XI Jornadas de Enseñanza Universitaria de la Informática, JENUI 2005*, Madrid (Spain), pp: 515-522, ISBN: 84-9732-421-8.

[3] Almeida, P., Viera Coito, F., Brito Palma, L. (2004) "An Environment for Remote Control". *1st International Workshop on e-learning and Virtual and Remote Laboratories*, VIRTUAL-LAB'2004, Setubal (Portugal).

[4] Bagnasco, A.; Chirico, M.; Scapolla, A.M. (2005) "XML Technology to Design Didactical Distributed Measurement Laboratory (RmwLAB) Instrument", *IEEE Transactions on Instrumentation and Measurement*, VOL. 54, Nº 1.

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

[6] Cabello, R. et al. (2004) "EMERGE: A European Educational Network for Dissemination of OnLine Laboratory Experiments". *Innovations 2004*, Ed. iNNER.

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

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

[9] García-Zubia, J et al. (2005) "A new approach for implementing remote laboratories: a practical case". *2nd International Symposium in Remote Engineering and Virtual Instrumentaion, REV 2005,* Brasov (Romania).

[10] Gomes, C. (2004) "Distance Learning Remote Laboratories using LabVIEW". *1st International Workshop on e-learning and Virtual and Remote Laboratories*, VIRTUAL-LAB'2004, Setubal (Portugal)

[11] Gomes, L.; Costa, A. (2004) "Embedded sytems introductory course supported by remote experiments ". *1st International Workshop on e-learning and Virtual and Remote Laboratories*, VIRTUAL-LAB'2004, Setubal (Portugal).

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

[13] Kahoraho Bukubiye, E., Larrauri Villamor, J.I. (2002) "A WebLab System for the Study of the Control and Protection of Electric Motors", *Proceedings of Telecommunication, Electronics and Control*, pp. 7. Cuba 2002. ISBN: 84-8138-506-2

[14] Ko, C.C.; Chen, Ben. M.; Chen, Jianping. (2004) *Creating Web-Based Laboratories*, Springer-Verlag

2004, London, ISBN: 1-85233-837-7
[15] Pelcz, A. et al. (2004) “Remote experiments using JAVA: Inplementations in the Virtual Electro Lab project". *1st International Workshop on e-learning and Virtual and Remote Laboratories*, VIRTUAL-LAB’2004, Setubal (Portugal).
[16] Soysal, O. (2000) "Computer Integrated Experimentation in Electrical Engineering Education over Distance" *Proceedings of ASEE 2000 Annual Conference*, Saint Louis, MO.