Developing a Second-Life-based Remote Lab over the WebLab-Deusto architecture

J. García-Zubia[1], J. Irurzun[1], I. Angulo[1], P. Orduña[2], J. Ruiz-de-Garibay[2], U. Hernández[1] and M. Castro[3]
[1] Faculty of Engineering, University of Deusto, Bilbao, Spain
[2] DeustoTech - Tecnológico Fundación Deusto, Bilbao, Spain
[3] IEECD - National University for Distance Education (UNED), Madrid, Spain

***Abstract*—Remote Labs didactic worth can be very supported using them to solve pressing educational problems. The contrasted lack of interest from young students for Engineering is a growing problem today for the Universities that will affect the global economies in the future. Awakening bachelor's interest for Science and Engineering is a possible way to restrain this tendency. Since different researches prove that the higher is the sense of presence in the environment the more effective becomes the learning process, using Remote Labs from highly immersive environments can be considered a promising idea. Following this educational aim, the present work covers the technical development of SecondLab, an experimental Remote Lab built over the WebLab-Deusto architecture and the popular 3D-based virtual environment Second Life.**

***Index Terms*—Remote experimentation, Second Life, Virtual Environments, e-learning**

I. Introduction

Remote Labs offer several well-known advantages both to be used as the main environment for hands-on teaching as well as to complement traditional laboratories [1].

On the other hand, different pedagogical research works have recently shown a growing disaffection for engineering among prospective students [2] [3] [4]. Related to this, new learning trends suggest entertainment and highly immersive environments as a way to achieve an effective learning process [5]. In particular, social-ware and game-like implementations are presented as attention-captive solutions applied to virtual education [6].

This work tries to contribute to fix the previous tendency by joining the well-known benefits from Remote Labs with the proved educational value from Virtual environments. Specifically, the present work covers the technical challenge of integrating both areas with different technologies than other people have done before.

The present paper is structured in three main sections. Section II first summarizes other previous works in this area. Section III shows an overview of the projects involved in the integration, while Section IV describes the implemented solution from a technical point of view. The research finishes in Section V with the conclusions and future plans.

II. Previous works

Using 3D game-like environments for educational purposes is not such a novel topic; the contribution is focused on the specific integration of real hardware (a Remote Lab) in a 3D game-like educational environment. Two projects have been recently published in this line.

A. UNIGRAZ, MIT and Curtin

A collaboration among the Graz University of Technology in Austria, the Massachusetts Institute of Technology in USA and the Curtin University of Technology in WA resulted in 2009 as the first Remote Lab accessible from a 3D virtual environment [7].

This research consisted on integrating three independent projects -Oracle/Sun Microsystem's Project Wonderland [8] as the 3D virtual environment, MIT's iLab [9] as the Remote Lab platform, and the MIT's Force On a Dipole Experiment as the experiment itself- to get a 3D-based collaborative environment from which to remotely practice with the experiment.

B. Bremen and UFRGS

After studying the general needs that every computer-supported collaborative social environment for education should cover [6], Dieter Müller from the University of Bremen in Germany and Frederico Menine Schaf from the Universidade Federal do Rio Grande do Sul in Brazil announced in 2009 [10] the current development of a 3D client under OpenSim [11] from which to use a PLC experiment based on a PROCESSING/WIRING-programmable [12] [13] Arduino.

III. Subprojects Involved

As previously explained, other research groups have already integrated Remote Labs with Virtual environments -iLab with Wonderland- or they are currently working on it -an “ad-hoc” Remote Lab server with OpenSim. The present work focuses on exploring the integration possibilities between two different projects to the previous ones: Second Life [14] and WebLab-Deusto [15]. Furthermore, a microbot has been chosen as the experiment to offer. The three underlying components are separately explained below.

A. WebLab-Deusto

WebLab-Deusto is the Remote Lab of the University of Deusto, which makes possible to offer real experiments to a certain group of users through any computer network, such as Internet. This Remote Lab has been used since February 2005 by the students of the University of Deusto as an essential tool for their practice works in four different engineering-related subjects.

The WebLab-Deusto architecture follows a distributed model (see Fig. 1) that makes easy to integrate new experiments inside, so every experiment served this way automatically takes advantage of all the common features implemented by WebLab-Deusto:

1) *Authentication*: WebLab-Deusto offers an extensible authentication system that currently supports three ways to verify the user credentials: username and password stored in a MySQL database, authentication through a remote LDAP server and trusted authentication based on the IP address of the client that is trying to log in.

2) *Reservations*: WebLab-Deusto offers a generic reservation system that makes possible to reserve any available experiment for a non-scheduled period of time (i.e: 200 seconds), which is defined as a customizable experiment-dependant property.

3) *Queue management*: WebLab-Deusto manages different reservation queues for the different available experiments, making the experiments themselves completely unaware of the users that are trying to use them.

4) *Scalability*: The WebLab-Deusto architecture is horizontally scalable since more hardware can be added to improve not just the performance but also the availability of the remotely served experiments. The more experiment hardware devices of the same type are added, the more users will be able to use the same experiment simultaneously.

5) *Security*: The WebLab-Deusto distributed architecture keeps in isolation the experiment-dependant hardware and software, so any problem related to a wrong use of the experiment will never put the whole Remote Lab at risk. Besides that, this same architecture considers the Login server separate from the others, so the most sensitive information (passwords) could not be accessed in case of eventual security attacks (see Fig. 2).

6) *Deployment*: The WebLab-Deusto deployment system makes easy and flexible to configure the distributed network map in which all the servers and experiments are involved. Every WebLab-Deusto server can be configured with an easy-readable .xml file to declare the protocols which it will be communicated through, being five different ones available up to now: XML-RPC, Python-dependant SOAP messages, TCP sockets, UNIX sockets and "Direct" (direct method calls inside the same process).

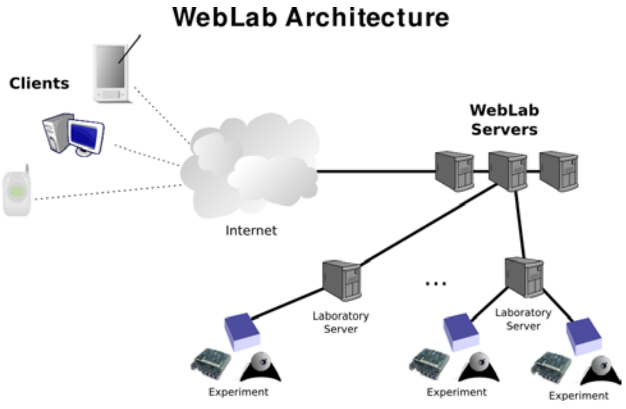

Figure 1: WebLab-Deusto architecture

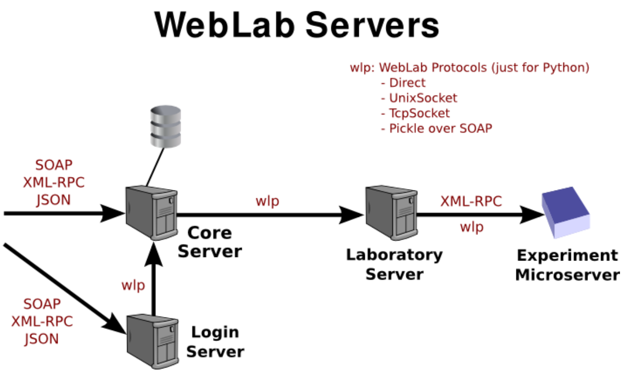

Figure 2: WebLab-Deusto servers

7) *Logging*: Everything that may happen during the use of the Remote Lab can be logged at different risk levels. The amount of events to log is up to the WebLab-Deusto administrator.

8) *Administration*: As a common feature that every Remote Lab requires, WebLab-Deusto offers administration tools that make easy both to monitor who is currently using the Remote Lab as well as to add, edit or remove users, permissions and experiments.

B. *Second Life*

Second Life is an end-user MMOW (Massive Multiplayer Online Worlds) software developed by Linden Lab since 1999. It provides a 3D environment where people can create their own customized avatar (a virtual representation of a person) and interact with other people. Its goal is to offer a real-world-like platform with the same physic laws, rules of conduct and opportunities (to get a job, to buy things…) that everybody has in the real life.

The most remarkable feature of Second Life is that there are no established goals for its users. Due to this, the platform has been mainly used in an entertainment way, but during the last few years it has also been used as an educational tool [16].

From a technical point of view, Second Life consists on two main software components: the private server farms at Linden Lab and the free-software client. The servers manage the business logic and the persistence tier, so the client is just responsible for rendering the 3D world and notifying the servers all the events carried out by the user. Therefore, developing software in Second Life entails both designing 3D graphics as well as programming server-side software.

C. *SecBot*

The SecBot experiment is an auto-programmable microbot specifically designed for this project. Despite that, it can be considered a background component, since any other experiment could have been offered in SecondLab instead (from a technical point of view) and the microbot itself could also be used in a different context.

Choosing a microbot as the experiment offered in Second-Lab is due to both educational and technical reasons. The educational one points to the attractiveness of the experiment for young students, which works in the direction of promoting entertainment while learning. The technical reason is related to the state-of-the-art in auto-programmable microcontroller memories. Now it's possible to create an autonomous microbot with the capability of programming its own instructions memory, so it can completely change its behavior in run-time. This is exactly the feature required to design the autonomous remotely-programmable microbot desired for the educational aims of the project.

SecBot has been built over the Robot Sumo Terminator S300230 from SuperRobótica, a commercial mechanical structure commonly used in the fighting modality at robotics competitions. As a first prototype version, SecBot offers the student the following components:

- 1x PIC18F4520 microcontroller from Microchip
- 1x MD22 engine controller circuit from Devantech
- 1x GP2D12 infrared sensor from Sharp

IV. Integration

SecondLab works as the glue code to make all the previous components work together. The best way to understand how SecondLab can be used is to look at the "happy path" scenario, described by the following steps:

1) Write a C program with the instructions that the microbot shall follow.
2) Compile the .c file to obtain the assembler .hex version of the program.
3) Open Second Life and log in with the avatar's name, surname and password.
4) Travel to SecondLab by typing "SecondLab" in the search tool.
5) Enter the laboratory with the avatar (see Fig. 3) and press the big green START button.
6) Upload the .hex program (see Fig. 4).
7) Wait until the screen in front shows the microbot working under the programmed orders (see Fig. 5).
8) Press the big red STOP button and leave the laboratory.

A. Overview

The SecondLab architecture is completely determined by the previously described background components. In particular, WebLab-Deusto determines the global architecture, as it is the Remote Lab platform underneath.

The WebLab-Deusto architecture follows an experiment-agnostic approach [17]; that is, it makes all its features completely independent of the specific requirements of the remotely-served experiments. Thus, integrating a new experiment in WebLab-Deusto becomes as simple as developing two components: a microserver at the very end of the server side (capable of interacting with the microbot), and a webclient module at the highest layer of the provided web-client application (presentation layer). Since the client platform had to be Second Life in this project, the most remarkable challenge became making Second Life work as a WebLab-Deusto client. Figure 6 shows the interaction among all these components.

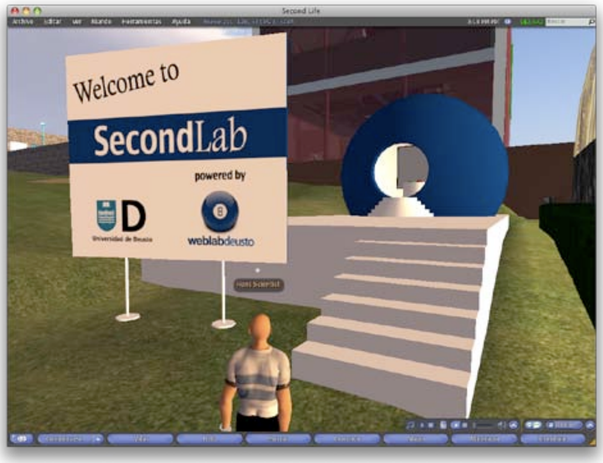

Figure 3: The SecondLab entrance

Finally, a third smaller component, Web-uploader, had to be developed too in order to solve a certain Second Life technical limitation that will be explained later.

B. Microserver component

A WebLab-Deusto microserver has to communicate the experiment-agnostic set of WebLab-Deusto servers with the final hardware of the experiment (the SecBot microbot in this project). The only requirement established by WebLab-Deusto to develop a microserver is to implement a short communication interface, which can be done in any of the following supported protocols: Direct (function calls from/to the same process), Pickle over UNIX Sockets (Python dependent), Pickle over Internet Sockets (Python dependent), Pickle over SOAP (Python dependent) and XML-RPC (language independent).

As figure 6 shows, the microserver in this project had not just to translate the messages received from WebLab-Deusto to the microbot, but also to manage with two different communication protocols: Bluetooth and any other from the five above. For this reason, a high-level and well-communicated language as Visual Basic .NET was chosen to develop this component. Therefore, XML-RPC was the only possibility to implement the listening interface.

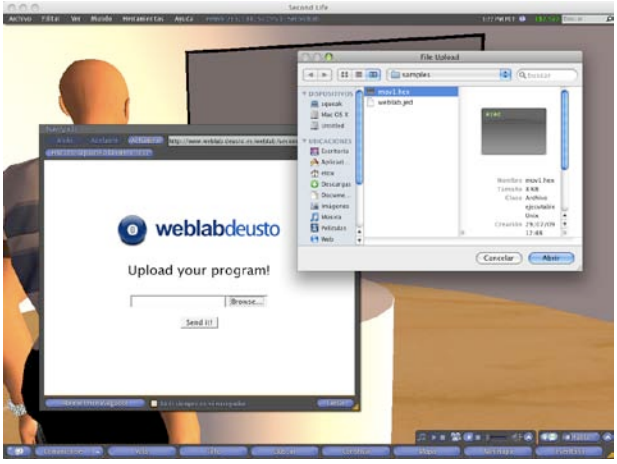

Figure 4: Uploading the program

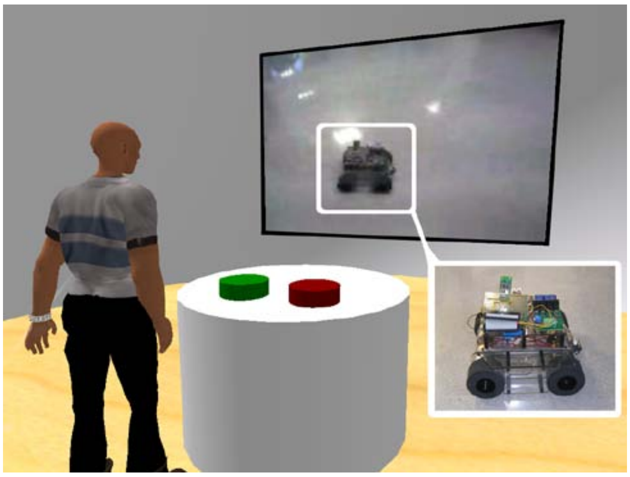

Figure 5: Watching the microbot working

C. *Client component*

The most significant challenge in this project was to make Second Life work as a WebLab-Deusto client. Since Second Life provides a generic world where people can live and build their lifes, it also provides technical features so developers can program the created objects and buildings with certain behaviors.

Developing a custom piece of virtual world in Second Life involves several steps. First, since building is not allowed for free, buying a plot of land is required in order to persist all the changes inside; secondly, every desired object must be graphically designed (buildings, items, textures…) using the 3D model editor provided by Second Life; and lastly, as many programmed behavior as wanted can be added to the world thanks to the scripting language created by Linden Lab (LSL, Linden Scripting Language). Every object in the world can be linked with many LSL scripts, which will be executed when the different programmed events happen in the virtual world (e.g.: an avatar touches a certain object).

From this scenario, developing a WebLab-Deusto client in Second Life entailed two big steps: designing a graphical 3D laboratory where the students could go with their avatars and communicating this laboratory with the Remote Lab so it could work exactly as a valid WebLab-Deusto client. Since the graphical task involved a creative work, the technical challenge was to develop a communication layer under the conditions of a limited language like LSL.

The WebLab-Deusto architecture presents the server side as a webservice with a public interface facade available in three different format protocols: SOAP, XML-RPC and JSON. In the other hand, Second Life offers both XML-RPC and the underlying HTTP protocols; however, the provided XMLRPC support is too limited to be used with a minimally complex service (only one integer and one string values can be retrieved at a time). Therefore, an XML-RPC layer has been manually implemented in LSL so the provided HTTP service is definitely used to communicate Second Life with WebLab-Deusto. This developed XML-RPC layer is limited to the specific use done in this project, so it cannot be considered a generic XML-RPC library for LSL. Although the original design considered it, the technical memory restrictions found in Second Life (64 KB/script including bytecode, stack and heap) made impossible that desirable approach.

Finally, as figure 6 shows, the communication between Second Life and the webcam that points to the microbot is done through the RTSP streaming protocol, thanks to the easy-to-use RTSP client support offered by Second Life, which only requires to specify the URL where the streaming is received from. The network camera used in SecondLab is the D-link DCS-5220, although any IP-based webcam with RTSP support could be used.

D. *Web-uploader component*

This web-server component covers a Second Life limitation. The LSL programming API does not offer any user interface component to allow the user to select a file from its computer. This shortage becomes a critical issue in a project like SecondLab, where sending a file from the student's computer is the main action in the interaction scenario.

Web-uploader takes advantage of the most flexible utility offered by the LSL programming API, which is a function that shows an embedded web browser with the specified URL loaded. The solution has been developed like a set of two server-side Python scripts. The first one offers a traditional HTML form with a file field that allows the user to choose a file from its hard drive (see Fig. 4). When the submit button is pressed, this script sends a HTTP POST request to the second one, which really sends the selected file to WebLab-Deusto and notifies Second Life that the file has been sent, so the web browser can be closed and the user can be told to look at the virtual screen in front.

V. Conclusions and Future work

SecondLab proves that it is possible to mix virtual environments with real experimentation in a first approach (prototype version). However, it also proves that developing a Remote Lab for end-users under Second Life is not recommended due to its serious technical limitations:

1) *Limited scripting language*. The Linden Scripting Language is not comparable with any modern programming language. Its most remarkable handicap is the limited support offered for data types, which makes impossible to manage minimally complex data structures in memory. Suffice it to say that LSL does not offer basic structures (struct in C) or dictionaries, being a very limited (non-nestable) list the most complex structure supported.
2) *Few User Interface components*. Second Life was not designed as a real development environment but as a platform where average-users could create buildings and interact with other avatars through simple question/answer dialogs. Therefore, the LSL programmer can only get input data from the user by two simple UI components: a dialog window with customizable answer-buttons (llDialog) and a text box where the user can write characters (llTextBox). As already explained, sending the user's .hex file in SecondLab had to be implemented using the embedded browser provided by Second Life in a quite unorthodox way (see Section IV).
3) *Memory restrictions*. Every LSL script is only provided with 64 KB of total memory (bytecode,

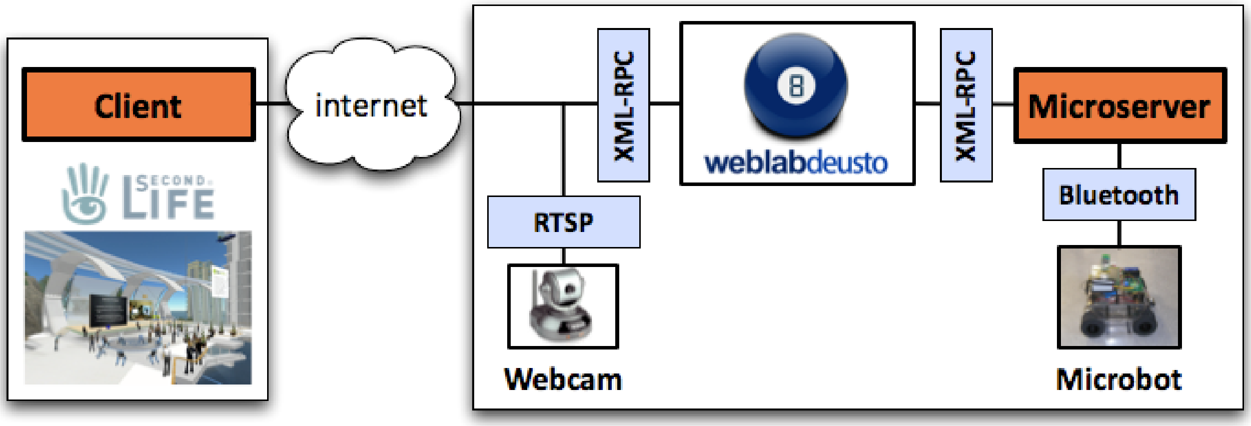

Figure 6: SecondLab architecture

stack and heap included) at the Linden Lab servers. Having checked during the development of this project that 1000 lines of LSL source code consume approximately 55-60 KB of memory, this restriction becomes a significant problem that makes Second Life a really limited environment where almost none ambitious project can be developed at. However, Linden Lab also enables to set up a personal server and connect it to their grid, so the environment restrictions imposed by the Linden Lab service might be avoided. Although SecondLab has not been deployed this way (since renting a land was enough for a prototype), this other approach may solve the memory limitations.

4) *Lack of high-level communication protocols*. As already said, although according to the documentation LSL offers both XML-RPC and HTTP as communication protocols, the provided XML-RPC support is too limited to be used with a minimally complex service. Therefore, HTTP becomes the only well-supported communication protocol, so any higher-level one must be manually implemented (as done with XML-RPC in this project).

According to these limitations, a production-ready 3D-based Remote Lab for end-users should consider the idea of choosing a different 3D environment, such as Open Wonderland [18] or OpenSim.

After getting around of the current Second Life issues, future plans for this project consider technical improvements both in the software and hardware sides.

The main software improvement will extend the functionality of the Remote Lab (based on Second Life or any other 3D environment) to offer two real microbots for two students at the same time, so different kinds of competitions may take part remotely in order to increase the students motivation.

In addition, the next SecBot version will rely on an auto-rechargeable battery system, so the microbot will be completely autonomous. This mechanism will be based on a circular customized ring. The walls of this ring will be filled with two conductor strips connected to the power supply, so as the microbot stops being used it can move forward and stay attached to any wall charging its battery.

ACKNOWLEDGMENT

The authors would like to acknowledge to the Spanish Science and Innovation Ministry for the support in the project TIN2008-06083-C03/TSI “s-Labs – Integración de Servicios Abiertos para Laboratorios Remotos y Virtuales Distribuidos”.

REFERENCES

[1] L. Gomes and J. García-Zubia, “Advances on remote laboratories and e-learning experiences”. University of Deusto, 2007.

[2] K.-K. Tan, K.-Z. Tang, V. Ng, A. Tay, S.-C. Yen, and T.-H. Lee, “Bridging physics to electronics: an outreach effort”. IEEE Transactions on Education, vol. 53, no. 1, pp. 3 –11, feb. 2010.

[3] A. Lopez-Martin, “Attracting prospective engineering students in the emerging european space for higher education”. IEEE Transactions on Education, vol. 53, no. 1, pp. 46 –52, feb. 2010.

[4] P. Molina-Gaudo, S. Baldassarri, M. Villarroya-Gaudo, and E. Cerezo, “Perception and intention in relation to engineering: A gendered study based on a one-day outreach activity”. IEEE Transactions on Education, vol. 53, no. 1, pp. 61 –70, feb. 2010.

[5] A. Cheney, S. Bronack, and R. Sanders, “Teaching and Learning in a 3D Immersive World: The AETZone Model”. Campus Technology, 2007.

[6] F. Schaf, D. Müller, C. Pereira, and F. Bruns, “Computer Supported Collaborative Social Environment for Education, Training and Work”. Remote Engineering and Virtual Instrumentation (REV), 2008.

[7] B. Scheucher, P. Bailey, C. Gütl, and J. Harward, “Collaborative Virtual 3D Environment for Internet-Accessible Physics Experiments”. International Journal of Online Engineering (iJOE), vol. 5, no. 5, 2009.

[8] Wonderland. [Online]. Available: http://www.projectwonderland.com

[9] MIT iLab. [Online]. Available: http://ilab.mit.edu

[10] D. Müller and F. Schaf, “A Low Cost Learning Environment for Collaborative Engineering”. Remote Engineering and Virtual Instrumentation (REV), 2009.

[11] OpenSim. [Online]. Available: http://opensimulator.org

[12] PROCESSING. [Online]. Available: http://www.processing.org

[13] WIRING. [Online]. Available: http://www.wiring.org.co

[14] Second Life. [Online]. Available: http://www.secondlife.com

[15] WebLab-Deusto. [Online]. Available: http://www.weblab.deusto.es

[16] T. Ritzema and B. Harris, “The use of Second Life for distance education”. Journal of Computing Sciences in Colleges, vol. 23, no. 6, pp. 110–116, 2008.

[17] P. Orduña, J. García-Zubia, J. Irurzun, E. Sancristobal, S. Martín, M. Castro, D. López-de-Ipiña, U. Hernández, I. Angulo, and J. González, “Designing Experiment Agnostic Remote

Laboratories," Remote Engineering and Virtual Instrumentation (REV), 2008.

[18] Open Wonderland. [Online]. Available: http://www.openwonderland.org

AUTHORS

Javier García-Zubia is with the University of Deusto, Electronics and Automation department, Avenida de las Universidades 24, 48007 Bilbao, Spain (e-mail: zubia@deusto.es).

Jaime Irurzun is with the University of Deusto, WebLab-Deusto Research Group, Avenida de las Universidades 24, 48007 Bilbao, Spain (e-mail: irurzun@deusto.es).

Ignacio Angulo is with the University of Deusto, Industrial Electronics, Control Engineering, and Computers Architecture department, Avenida de las Universidades 24, 48007 Bilbao, Spain (e-mail: ignacio.angulo@deusto.es).

Pablo Orduña is with DeustoTech - Tecnológico Fundación Deusto, Ambient Intelligence department, Avenida de las Universidades 24, 48007 Bilbao, Spain (e-mail: pablo.orduna@deusto.es).

Jonathan Ruiz-de-Garibay is with DeustoTech – Tecnológico Fundación Deusto, Ambient Intelligence department, Avenida de las Universidades 24, 48007 Bilbao, Spain (e-mail: jonathan.garibay@deusto.es).

Unai Hernández is with the University of Deusto, Telecomunications department, Avenida de las Universidades 24, 48007 Bilbao, Spain (e-mail: unai.hernandez@deusto.es).

Manuel Castro is with the National University for Distance Education (UNED), Electrical and Computer Engineering department, C/ Bravo Murillo 38, 7a Planta, 28015 Madrid, Spain (e-mail: mcastro@ieec.uned.es).

This work was supported in part by the Regional Government of the Basque Country (project SAIOTEK S-PE08FD03).