

# Remote Laboratory in Education: WebLab-Deusto Practice

Olga Dziabenko<sup>1</sup>, Pablo Orduña<sup>1</sup>, Javier García-Zubía<sup>2</sup>, Ignacio Angulo<sup>2</sup>

<sup>1</sup> DeustoTech, <sup>2</sup> Faculty of Engineering

University of Deusto

Spain

olga.dziabenko@deusto.es, pablo.orduna@deusto.es, zubia@deusto.es, ignacio.angulo@deusto.es

**Abstract:** This paper describes the current influence of the remote laboratory on practical learning aspects of high education. The key challenges facing the teaching of science and engineering include insufficient hands-on laboratory usage in classrooms. The main objective of the paper is to present different learning approaches, supported by the remote lab WebLab-Deusto and the principal structures of existing remote laboratories: iLAB (MIT, USA), Labshare (UTS, Australia), VISIR (BTH, Sweden). Educators can benefit from different teaching methods (3D collaborative, game-, inquiry-, and project-based learning) integrated in WebLab-Deusto.

## Introduction

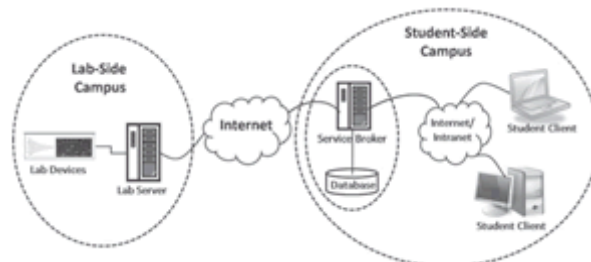
To the question “why has a remote laboratory been designed for your faculty,” the usual answers are “to improve laboratory workload for students with engineering or physics degrees” or “to look for new and efficient forms of education”. In 2005, as MIT Professor, Jesus del Alamo said “If you can't come to the lab...the lab will come to you!” The key challenges facing STEM (Science, Technology, Engineering, and Mathematics) education include insufficient usage of laboratories, hands-on and/or remote, in classroom. It is accepted that traditional laboratory experimentation is a crucial part of lecturing STEM students, one that provides principal knowledge with real, physical experiments. In the economic situation where education institutions had limited resources to purchase the expensive laboratory equipment, the remote laboratory became a unique solution for providing students with such experience and nowadays industrial instrumentation. Historically, the development of remote experiments is directly connected to the appearance and progress of the Internet. The first remote laboratories were mentioned in 1994-1995 by teams at the Oregon Joint Graduate Schools of Engineering (Bohus 1995, 1996) and at the University of Tennessee at Chattanooga (Jim,1996). When the Internet became available for everybody, remote laboratories started to provide students with an enhanced way to learn about physical systems, interact with them, to repeat experiment multiple times, and to share, collaborate and supplement engineering courses and experiments with faculty and students throughout the world. The combination of real systems, high-bandwidth data connections, augmented reality technology and effective pedagogic design can improve students' ability to understand the nature of the physical processes. In Chapters 1-3, this paper provides a description of different remote labs: iLab (MIT, USA), Labshare (UTS, Australia), VISIR (BTH, Sweden). Chapter 4 presents a summary of the WebLab-Deusto remote laboratory with a variety of approaches such as game concept, Second Life, and m-learning issues. The conclusion offers several thoughts about the future of the WebLab-Deusto remote laboratory in education.

## iLab (MIT, USA)

In 1998 the first remote experiments were developed in the frame of iLab. The main goal of the iLab project was to define a “standard approach to the development of online labs” and to give “tools to make such development simpler for those wanting to create new labs” (DeLong, 2011). iLabs provide a “unique scaling properties and potentially enormous economic advantages with revolutionary impact in science and engineering education at all levels”. As result, a distributed service infrastructure - the iLab Shared Architecture - was developed during the iLab Project.

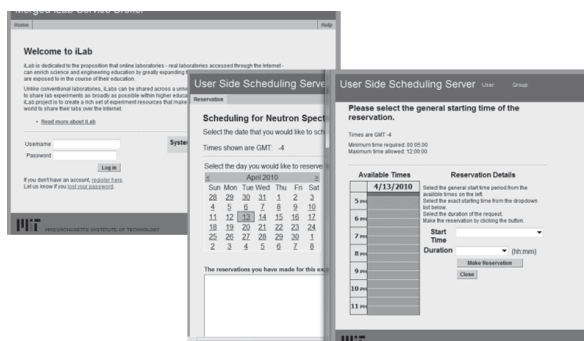
The iLab Project at MIT provides a common remote lab architecture (Fig.1) for the development and deployment of online laboratories titled the iLab Shared Architecture (ISA) (Harward, 2008). ISA is a robust,

scalable, open-source infrastructure. It is created on web services that provide a unifying software framework to support access to a wide variety of online laboratories. The ISA provides generic lab services in a middleware system over web services, such as user account management, scheduling and data storage. The main ISA consists of three main components: Service Broker, Lab Server, and Lab Client (Hardison, 2008). The Service Broker is responsible for authentication, authorization, experiment storage, user management, and management of the lab experiment on the lab server. The Service Broker can access multiple lab servers potentially located at multiple remote campuses anywhere on the Internet.



**Figure 1:** ISA components: Service Broker, Lab Server and Lab Client.

The Lab Server ensures independent administration of the lab equipment and maintenance of the experiment hardware. Since each lab server is specific to its experiment hardware, a different lab server must be used for such lab instrumentation. The ISA supports multiple Lab Clients that utilize the same backend Lab Server and experiment hardware.



**Figure 2:** Student view of the iLab Service Broker: scheduling pages, the login and experiment launch screen.

The usage of remote laboratory experiment from the student perspective involves three steps: scheduling time on the experiment hardware, running the experiments and collecting data, and downloading and analyzing their results (Fig.2). In addition to the student perspective, the faculty members have extra privileges which allow them to run and download reports, check user login information and see student experiment results. Most of the users of remote experiments are MIT undergraduate students in the departments of Nuclear Science and Engineering (NSE) and Physics. Students typically work in teams. The students doing the lab as part of a required course used the on-line option. It is clear that the benefit of online access is that the students “accumulate counting data over much longer times” (DeLong, 2011) and obtain better quality and quantity of data sets. Students can repeat experiments if data anomalies are discovered.

## Labshare - Sahara Architecture (UTS, Australia)

Sahara is the redeveloped system as a part of the Labshare project in order to design highly flexible and sophisticated remote laboratory system by extending the functionality, and creating a more maintainable and extensible architecture. It was developed by the Faculty of Engineering and Information Technology at the University of Technology, Sydney (UTS) (Labshare Institute, 2011). The early UTS remote laboratory system dates to the period 2000-2005 for flexible access of students to limited laboratory resources (Murray, 2008).

The flexible authentication and access from different institutions in order to share rigs and other attributes are essential features of the Sahara. Queuing, booking and user management functions are independently integrated on the rigs. When new rigs are added no additional development is required. The supplementary features of Sahara include: multiple rig operating modes, maximizing rig usage, and “agnostic to rig” (Lowe, 2011).

The core architecture of Sahara consists of the Scheduling Server, the Web Interface Client and the RigClient. The architecture is based on SOAP (Simple Object Access Protocol) interfaces, using extensible markup language (XML) for the means of communication. This architecture has been designed to “facilitate short rig design time and low costs in creating a new experimental rig” (Lasky, 2007; Murray, 2006). Later versions of Sahara have progressively improved to avoid various limitations that became evident through large scale usage.

Sahara is based on client-server architecture. The core of Sahara is the Scheduling Server with a stack, the bottom of which manages the rig clients and the queuing. At the top, the session creation and management is handled. Sahara provides generic functionality - independent of specific rigs and rig types. For this purpose any rig integrated into Sahara needs a “rig client” offering an interface between the hardware and the Sahara system. There are three main types of rig clients which are based on the user control interface interaction with the rig. The options for rig control are:

- peripheral control - the control interface is ‘outside’ of Sahara. In this case, the Sahara rig client has no direct control of the hardware, but is responsible for assigning access to the control program to the user, revoking access, starting and monitoring test results, and detecting activity on the rig. An experiment that already works through a computer interface, e.g. using a control program such as LabView, though not remotely, would be a typical implementation of this rig type to adapt to a remote laboratory.
- direct or primitive control – directly controlled by the rig client. The rig client is responsible for controlling the rig hardware to perform the experiment. This is suitable for security restrictions and where there is no existing control application developed, or where low latency for graphical elements is not essential.
- batch control - where user interaction is not required for the laboratory to be completed. The rig client is responsible for loading the control file, for verifying its contents and for parsing it so that the appropriate action is taken on the rig.

The chosen rig type depends on the rig itself and the type of experiments and lessons required for the rig. Many factors can influence on this choice such as control richness, time consuming, flexibility, etc. Although examples of remote labs used in university coursework subjects can be increasingly found all over the world, many of them consist of one type of experiment only, and/or single rig instances. The Labshare provide a solution of this aspect by effective developing, utilization and sharing of laboratories to strengthening practical laboratory education.

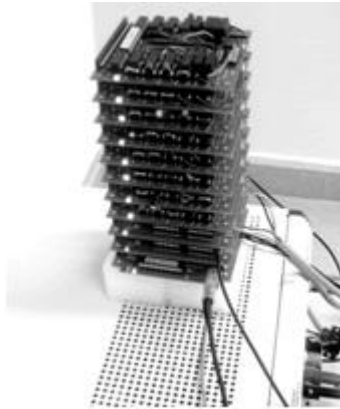
## **VISIR Open Lab Platform (BTH, Sweden)**

Electrical experiments are common at schools and universities worldwide. Usually students execute experiments in the frame of traditional hands-on laboratories. The existing remote laboratories are more or less replicas of hands-on ones. Virtual Instrument Systems in Reality (VISIR) Open Lab Platform is a remote laboratory created by Blekinge Institute of technology (BTH) for designing, wiring and measurement of electronic circuits. A student creates circuit using instruments and wires on his “PC-screen”. Then, VISIR Open Lab Platform converts the student’s design to a real wired circuit and sends the measurement results to the PC-screen. Thus, VISIR Open Lab Platform is a real electronic lab environment which can be accessed at any time and from anywhere over an internet (Gustavsson, 2009).

There are three main scenarios for practical learning using VISIR Open Lab Platform (Feisel, 2005):

- (1) High schools and freshman university students. They can use VISIR Open Lab Platform for performing basic DC and low frequency AC circuits experiments related to Ohm’s and Kirchhoff’s laws. Moreover, the students will become familiar with instruments, components, manuals, data sheets, circuit wiring, and other laboratory work.
- (2) University students who are familiar with components, circuit wiring and performing on complicated circuits experiments including transistors, operational amplifiers, AD converters, etc.
- (3) Experiments on fixed circuits, e.g., on printed circuit boards.

The main part of VISIR Open Lab Platform architecture is online workbench (Fig.3) that is similar to hands-on laboratories at any school or university. Workbench for electrical experiments contains power sources, measuring instruments and a solderless breadboard. A relay switching matrix helps to assemble circuits remotely. It integrates connectors, sockets for components, and relays into a matrix pattern – in rows and columns. The VISIR Open Lab Platform provides a virtual component box and virtual breadboard for controlling the switching matrix and for “wiring” circuits remotely with the mouse pointer (Gustavsson, 2008). Remotely the teacher selects components using the button “+” from the online component store and puts them into the component box. A virtual instructor was designed and developed to protect the online workbench from student’s harmful errors and damages during experiments.



**Figure 3:** The online VISIR workbench at WebLab - Deusto, University of Deusto.

The VISIR Open Lab Platform offers virtual front panels of the desktop instruments and the multi-meter in image/photo format. Adobe Flash animations enable students to turn the knobs and buttons, and make settings using the mouse pointer. In order to avoid misunderstanding and provide immersion of the remote experiment, the virtual panels should be an exact virtual copy of devices which students use at hands-on laboratories if applicable. A single VISIR Open Lab Platform workbench can emulate a laboratory with many workbenches (Gustavsson, 2008). However, the experiment sessions, from a pedagogical approach, should be organized as in a hands-on laboratory - one workbench per student or student team (Gustavsson, 2011). Each remote student or student team specifies the experiment locally on their own computer and sends a message containing a description of the desired circuit and the instrument settings to the VISIR Open Lab Platform workbench (server). The experiment result or an error message is returned to the requesting client computer after performing the procedure. In case of server occupation, the experiment request is queued. The circuit creation and measurement procedure should be set to 0.1 seconds to get a reasonable response time even with a large number of experimenters. Otherwise, more hardware is required to reduce time waiting in queues and improve experiment duration.

### **WebLab-Deusto (UDeusto, Spain)**

Since 2005 the Faculty of Engineering of the University of Deusto has provided WebLab-Deusto - a remote laboratory for technical curricula. WebLab-Deusto is focused on the development of an infrastructure that enables students to access a hardware of experiments located at the University over the Internet. Today’s WebLab-Deusto 4.0M1 is a robust platform that uses web standards suitable for mainstream web browsers and adapts to mobile devices. It can be downloaded and deployed to serve new remote experiments in different environments and operating systems. The general features of the system are a secure and scalable design, using SSL in the communication, checking experiment in the management layers, and tracking down the use of the system and the messages sent by students. The system also supports different authentication schemes such as a regular database, LDAP, supported external students OpenID. It is extensible enough to be integrated into a number of platforms such as Facebook, .LRN and Moodle, and to integrate external experiments such as VISIR.

## WebLab-Deusto Architecture

Currently the experiment can be developed using two different approaches: managed and unmanaged experiments (Orduña, 2011).

### *Managed experiments*

WebLab-Deusto provides a Secure Sockets Layer to avoid the experiment code to handle encryption issues and unauthorized client performance. These software components are executed in the middle between client and server and handle communications. Since all communications - security and tracking - are fully managed by WebLab-Deusto, this type of experiment is called “managed”. FPGA experiment is an example of “managed” experiment (Fig.4):



**Figure 4:** WebLab-Deusto; left – “managed” experiment: FPGA, right – “unmanaged” one: Virtual Machine.

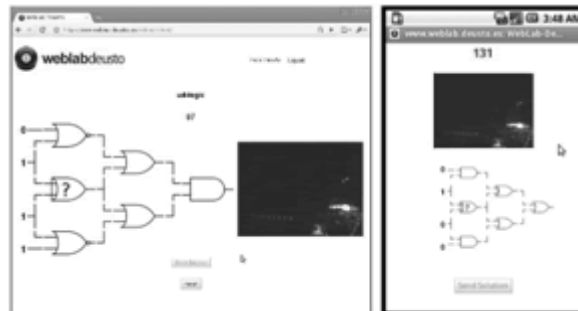
### *Unmanaged experiments*

WebLab-Deusto also supports “unmanaged” experiments for developers who are experts in the domain of the experiments but not in web development. While the authentication, authorization, scheduling, load balancing and basic user tracking are still handled by the platform, a client contacts the server on its own, without relying on the WebLab-Deusto libraries. Figure 6 shows “unmanaged” experiments - a Virtual Machine running a LabVIEW experiment (Fig.4).

### *Remote Experiments for Mobile devices*

Because mobile devices are popular among college students an adaptation of remote laboratories to mobile platforms such as iPhone and/or Android was done. In December 2006, the WebLab-Deusto 2.0 presented access from Symbian mobile devices with the Opera Mobile Web browser (López-de-Ipiña, 2006).

Currently WebLab-Deusto provides two versions of experiment-independent user interface: desktop and mobile versions. An experiment ud-logic has been adapted to the mobile version (Fig. 5) by using smaller images and reorganizing the layout of the experiment (Orduña, 2011).



**Figure 5:** WebLab-Deusto; left - the desktop version, right - the mobile version running on an Android 1.6 emulator.

Offering remote experiments for each mobile platform is expensive in terms of development and

maintenance. All versions of experiments should be updated every time a new feature of mobile device appears. Google Mail provides a mobile web interface, which is available only on Android, Blackberry and Symbian, while it relies on a web version for iPhone, Windows Mobile, and other mobile devices. Also Facebook provides a mobile web version supported by most mobile phones, but it requires more integration development with the mobile device. For example, Google Mail can use a regular mail client to receive notifications while Facebook needs extra coding to be mapped to other protocols. Therefore, the decision of implementing the native or the web version will rely on the demand of experiment for the most common mobile platforms used by students over the workload and budget consuming. This decision should be made for every experiment independently.

### WebLab-Deusto approaches

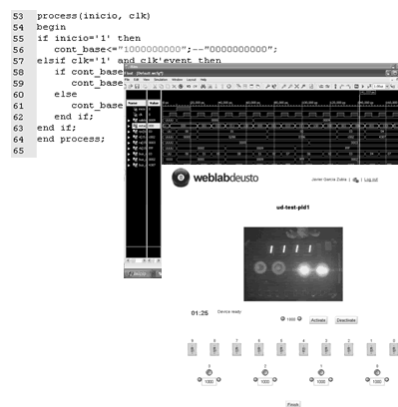
WebLab-Deusto pools their resources for the development of such scenarios where students access hardware of experiments located at the University electronically. Seven types of different experiments are offered by WebLab-Deusto:

- Xilinx Programmable Logic Devices;
- Microchip PIC Microcontrollers;
- Programmable Mobile Robot;
- Electronics & instrumentation;
- Logic games and custom C++ application connected to an oscilloscope through GPIB;
- Weblab-Submarine: a real aquarium is remotely maintained.

### Xilinx Programmable Logic devices

Xilinx Programmable Logic devices are long-term running experiments offered by WebLab-Deusto. Two experiments are based respectively on the family of Xilinx CPLD and XC95XX Spartan 3 FPGA. These experiments have been used in various subjects such as "Computer Structure" and "Programmable Logic" that are offered for third-year students of Electronics and Control Engineering Degree. The learning objective is to teach design with programmable devices. The course is based on the project-based learning approach offering students the labs assignments and projects as main activities.

The students learn to program in VHDL, and to develop, implement and test their design on an educational board in a hands-on laboratory. The assignment includes VHDL code writing, its simulating, and obtaining a binary file (Fig.7). Students access WebLab-Deusto - a demo is available - using the university authorization validated through LDAP. Once logged-in, they upload the generated binary file. After uploading generated binary file students control the board during 200 seconds. The board, identical to that used in the classroom laboratory controls 6 LEDs and 4 seven-segments, allowing the student to act on 10 switches, 4 buttons and a programmable clock from web interface. The behavior of designed system is observed over WebCam (Fig.6).

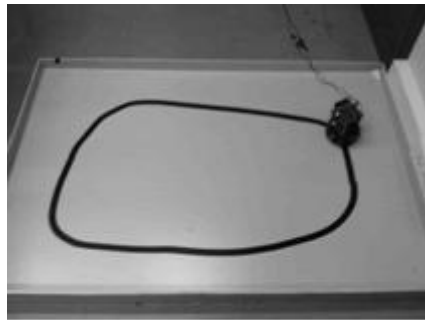


**Figure 6:** Xilinx CPLD device; VHDL code writing, simulating, and WebCam observing.

### Microcontrollers

During E-Pragmatic EU project an online training on Microcontrollers was designed. The training consists of "Introduction to Microcontrollers", "8-bit Microcontrollers Advanced Course", and "Low-cost

platform to provide LAN / WAN connectivity for embedded systems”. These courses cover different families of 8-bit microcontrollers manufactured at Microchip Technology Inc. (Arizona). Nowadays microcontrollers allow the automation of simple systems by performing simple processes responsible for automatic control of one or more actuators based on the capture of certain sensors. Current 8-bit microcontrollers apply from simple projects to complex ones - systems control including the latest industrial technologies of communication, at low cost and with minimal power consumption. The courses that are offered by WebLab-Deusto for learning PIC microcontrollers include three remote experiments. They manage: basic resources (digital inputs / outputs, timers, watchdog, etc.) in introductory course; complex peripherals such as PWM, ADC, Priority Interrupts, SPI and I2C buses, etc. in course of advanced peripherals and telecontrol; and an embedded system from internet with Ethernet connectivity in last one.



**Figure 7:** Microcontroller; Introduction to Microcontrollers course.

### ***Programmable mobile robot***

The course aims to encourage student interest in STEM through the use of two technological resources: virtual 3D environments (Second Life) and mobile robot programming. The remote labs developers actively use a 3D virtual world, such as Second Life, OpenSim or Open Wonderland for the presentation of real hardware experiments there. It is worth mentioning the work of Scheucher et al for integrating iLab “Force On a Dipole Experiment” into Open Wonderland (Scheucher, 2009) to get a 3D collaborative environment for remotely practice, and Müller et al (Müller, 2009) for the development of a 3D client under OpenSim where a PLC experiment based on a processing/wiring programmable Arduino is executed.

The team WebLab Deusto used Second Life for merging SecBot experiment (García-Zubia, 2010). The SecBot experiment is an auto-programmable microbot which is designed for this project. The SecBot microbot has its own programming instructions memory that allows completely changing its behavior in run-time. The SecBot has been built over the TERMINATOR SUMO ROBOT KIT by Lynxmotion managed by an embedded platform based in a PIC18F4520 microcontroller.

Design the remote experiment in Second Life includes a development of a project with the instructions that the microbot shall follow; high level compilers (e.g. MPLAB-C) or graphical programming interfaces (e.g. flowcode) should be applied; and uploading the .hex program and run it over WebLab-Deusto. Screen will show microbot movement under the programmed orders (Fig.8).



**Figure 8:** Second Lab; microbot movement under programmed orders.

Although the project was successful, several restrictions have influenced on the presentation of the experiment such as the Linden Scripting Language applying at Second Life is not comparable with any modern

programming language; few User Interface components; memory restrictions - each LSL script is provided with only 64 KB of total memory; lack of high-level communication protocols.

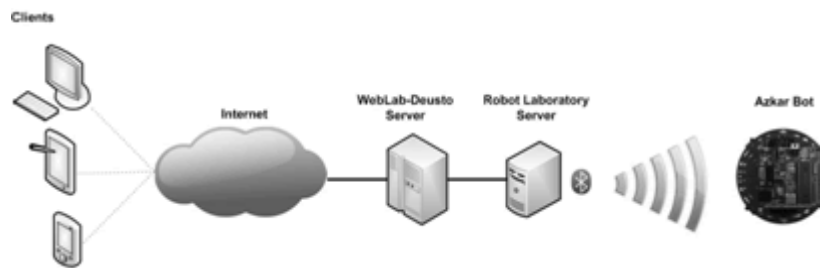
### Game Design

The primary goal of WebLab-Deusto game developers is to offer educators with an online product, the featured robotics tools ranging from all stages of exercises including robot programming to single lesson-play puzzles that can be easily incorporated into classroom activities. We assume this education tool will inspire participation in robotics educational programs and pique students' interest in STEM.

*During the landing on Triangulum Galaxy planet an environmental robot Max375v.11075 was crashed and his parts were spread out on a surface of the planet. The task of the players is to collect and assemble the Max's parts as fast they can in order to avoid his damages, using a strategy, and different game action levels, options and challenges.*

The robot design was carried out by trying to develop a device that meets the requirements demanded by the nature of game design and of the remote experiment: reduce cost, minimize user waiting times, and deploy redundant experiments by different institutions. The ROBOT has been built over AzkarBot (IMP, 2012). This product allows students to write code for a robot, experiment with and test it.

To create a good educational game using remote laboratory experiments the basic steps of game design, remote laboratory construction of experiments, elements of learning, motivation, and engagement outlined below should be taken into consideration: determine pedagogical approach (what model you will use, how students will learn); situate the task; elaborate the details; incorporate underlying pedagogical support; map learning activities to interface actions; map learning concepts to interface objects; analyze the technical limitations of the remote experiments; create relevant storytelling; design game balance – balance of challenges and motivation; optimize cognitive load - avoid irrelevant multimedia elements, provide friendly and usable user interface, and apply the modality effect.



**Figure 9:** AzkarBot architecture under WebLab-Deusto Platform.

There are many interactive learning techniques that have already been used in game-based learning. One of those techniques is “inquiry-based learning”, where execution of the remote experiments will teach students that good hypotheses for easy problem may lead to hypotheses that work well for harder problems (Skadberg, 2004) in order to find creative solutions to complex problems; and think about relationships of facts and actions. The open-ended structures of the educational game lets learners be producers by allowing them to modify or even design new scenarios.

Using this game, students can understand and learn how to manipulate the equipment remotely, what technical limitations exist, and how to reduce or even avoid it. The open-content structure and tips & tricks feature of the game make it applicable for different school subjects, such as foreign language, chemistry, mathematics, technology, etc. (Dziabenko, 2011) The teacher can design his own teaching content and its level of complication depends on the objectives of the curriculum. Currently, the interface of the game is in the developing stage while the remote experiment is ready for integration.

### Conclusion

In this paper we discussed the aspect of integration of remote laboratories into the curriculum. Thanks to contemporary communication instruments, common resources including remote experiments are available for



students worldwide. That is a great solution to illuminate the challenges and meet the demands of growing globalization in engineering education.

Our future work can be presented in three aspects: enhancing the service of remote laboratories for higher education, expand remote experiments for the secondary school STEM curriculum and promote it for science museum exhibition as a unique informal e-learning tool for emphasizing hands-on experience. For our first activity we are integrating our labs with other lab brokers, such as iLabs at MIT in frame of the iLabs-Europe, Labshare, and VISIR; we are developing more experimental modules; we are supporting live video and audio to our experiments that do not have it; we are personalizing the web interface of the remote laboratories and experiments; we are improving our equipment information for student users; we are presenting some of our remote systems on mobile devices OS such as Android, Symbian, and iPhone; we are presenting WebLab-Deusto on different learning environment –Facebook, Moodle, and Graaasp. The second aspect includes a design and development of three learning experiments for secondary schools teachers. We are creating the Weblab-Submarine, which allows one to remotely maintain an aquarium and control a movement of toy-submarine in it. Moreover, WebLab-Deusto manipulates real hardware in Baratzte farm from classroom in Urdaneta School. All three places are in approximately 15km from each other. This shows that the hardware of experiment should not be close to the remote laboratory. Finally, in the frame of the EU project OLAREX we are designing an exhibition at National Polytechnic Museum, Sofia (BG), where different remote laboratory approaches will be promoted.

All these future activities require a close collaboration between remote laboratories researchers and representatives of education in our society - teachers, trainers, lecturers, museum employees, and educators - for the facilitation of the learning experience in the remote experimentation setting.

## Acknowledgements

This work was supported by Lifelong Learning Programme of the European Union within Leonardo da Vinci (E-Pragmatic project 510586-LLP-2010-SI-LNW) and Key Activity 3 - ICT (OLAREX project 518987-LLP-1-2011-1-ES -KA3-KA3MP). However, any opinions, findings, conclusions, and/or recommendations are those of the investigators and do not necessarily reflect the views of the European Commission.

## References

- Bohus, C., Crowl, L. A., Aktan, B., and Shor, M.H. ( 1996) “Running Control Engineering Experiments Over the Internet”. *Proceedings of the IFAC World Congress*, San Francisco, California, vol. G, pp. 25-33.
- DeLong,K., Harward,J., Bailey, P, Kohse,G (2011). “Online Spectrometer Experiments Using the iLab Shared Architecture”. In García-Zubía *et al*(Eds.), *Using Remote Labs in Education: Two Little Ducks in Remote Experimentation* (pp. 121-134). Bilbao: University of Deusto.
- Dziabenko, O., García-Zubia, J. (2011)”Remote Experiments and Online Games: How to Merge them?” *International Journal of Engineering Pedagogy (iJEP)*, Vol 1, No1.
- Dziabenko, O., García-Zubia, J., Angulo, I., (2012) ”Time to play with a microcontroller managed mobile bot” *Proceedings of the EDUCON 2012 Conference*. Marrakesh, Morocco.
- Feisel, L.D., Rosa, A. J.( 2005) “The Role of the Laboratory in Undergraduate Engineering Education”, *Journal of Engineering Education*, pp 121-130.
- García-Zubia,J. , Irurzun,J., Angulo,I., Orduña,P., Ruiz-de-Garibay,J., Hernández,U., Castro,M. (2010). ”Developing a Second-Life-based Remote Lab over the WebLab-Deusto architecture”. *Proceedings of the REV 2010 Conference*. Stockholm, Sweden.
- García-Zubia,J., Gustavsson,I., Orduña,P., Lopez-de-Ipina, D., Hernández,U., Angulo,I., Dziabenko,O.,
- Rodriguez-Gil,L.(2011) “Using VISIR at the University of Deusto: experiments, subjects and students” *Proceedings of the REV 2011 Conference*, Brasov, Romania.

Gustavsson, I. *et al.*, (2008) "A Flexible Electronics Laboratory with Local and Remote Workbenches in a Grid", *International Journal of Online Engineering*, Vol. 4, No. 2.

Gustavsson, I., Nilsson, K., Zackrisson, J., García-Zubía, J., Hernández-Jayo, U., Nafalski, A., Nedic, Z., Göl, Ö., Machotka, J., Pettersson, M. I., Lagö, T. and L. Håkansson, (2009) "On objectives of instructional laboratories, individual assessment, and use of collaborative remote laboratories", *IEEE Transactions on Learning Technologies*, 2(4), pp.263-274.

Gustavsson, I., Alves, G., Costa, R., Nilsson, K., Zackrisson, J., Hernandez-Jayo, U., Garcia-Zubia, J. (2011) "The VISIR Open Lab Platform 5.0 - an architecture for a federation of remote laboratories", *Proceedings of the REV 2011 Conference*, Brasov, Romania, June 28 – July 1, 2011.

Hardison, J., DeLong, K., Bailey, P., Harward, V.J., (2008) "Deploying Interactive Remote Labs Using the iLab Shared Architecture," *Frontiers in Education (FIE) Conference*, Saratoga Springs, New York.

Harward, V.J., *et al.*, (2008) "The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories," *Proceedings of the IEEE*, vol. 96, no. 6, pp. 931-950.

Henry, J. (1996), "Controls Laboratory Teaching via the World Wide Web." *ASEE Annual Meeting*, Washington, D.C. <http://chem.engr.utc.edu/asee/ASEE-96-full.html>

Lasky, V. L., & Murray, S. J. (2007). "Implementing viable remote laboratories using server virtualisation". In V. Uskov (Ed.), *Web-based Education* (pp. 68-72). Chamonix, France: ACTA Press Anaheim, CA, USA.

López-de-Ipiña, D., García-Zubia, J., & Orduña P. (2006) "Remote Control of Web 2.0-enabled Laboratories from Mobile Devices" *Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing* pp. 123-123.

Lowe, D., Machet, T., Kostulski, T., (2011). "UTS Remote Labs, Labshare, and the Sahara Architecture". In García-Zubía *et al.*(Eds.), *Using Remote Labs in Education: Two Little Ducks in Remote Experimentation* (pp. 403-425). Bilbao: University of Deusto.

Murray, S. J., & Lasky, V. L. (2006). "A Remotely Accessible Embedded Systems Laboratory". In Sarkar (Ed.), *Tools for Teaching Computer Networking and Hardware Concepts* (pp. 284-302). Hershey: Information Science Publishing.

Murray, S., Lowe, D., Lindsay, E., Lasky, V., & Liu, D. (2008). "Experiences with a Hybrid Architecture for Remote Laboratories". *FiE 2008: The 38<sup>th</sup> Annual Frontiers in Education Conference*. Saratoga Springs, USA.

Müller, D., Schaf, F. M. (2009). "A Low Cost Learning Environment for Collaborative Engineering". *Proc. 6th International Conference on Remote Engineering and Virtual Instrumentation (REV'09)*. Bridgeport, USA.

Orduña, P., Irurzun, J., Rodriguez-Gil, L., Garcia-Zubia, J., Lopez-de-Ipiña, D., (2011) "Reusing requirements among remote experiments for their development and integration under WebLab-Deusto" *Proceedings of the REV 2011 Conference*, Brasov, Romania.

Orduña, P., Garcia-Zubia, J., Irurzun, J., Lopez-de-Ipiña, D., Rodriguez-Gil, L. (2011) "Enabling mobile access to Remote Laboratories" *IEEE EDUCON 2011 (ISBN: 978-1-61284-641-5)*.

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

Skadberg, Y. X., Kimmel, J. R. (2004) "Visitors' flow experience while browsing a web site: its measurement, contributing factors, and consequences," *Computers in Human Behavior*, vol.20, pp.403-422.

The Labshare Institute, "Remote Labs: Enriching digital education." <http://www.labshare.edu.au/>.