# Exploring students collaboration in remote laboratory infrastructures

Pablo Orduña[1], Luis Rodriguez-Gil [2], Ignacio Angulo[1], Olga Dziabenko[1], Diego López-de-Ipiña[1], Javier García-Zubia[2]

[1] Deusto Institute of Technology – DeustoTech, University of Deusto, Bilbao, Spain

[2] University of Deusto, Bilbao, Spain

***Abstract*—Educational remote laboratories are a software and hardware tool that allows students to remotely access real equipment located in the university as if they were in a hands-on-lab session. Since the equipment used by students is real, it has associated costs: laboratory development, hardware used and maintenance costs. Given the remote nature of the remote laboratories, institutions can share these costs by sharing the access to the laboratories. In order to reduce the associated development and maintenance costs, as well as to reduce the overall costs by managing the sharing of laboratories in different institutions, software infrastructures and toolkits have arisen, such as the MIT iLab project, the Labshare Sahara project, or WebLab-Deusto. However, a particular feature seamlessly present on hands-on-lab sessions but not often present in remote laboratories sessions is direct collaboration among students. While collaboration at a particular laboratory level is generally supported –or can easily be implemented-, some features of remote laboratory management systems such as load balancing or federation might enter in conflict with collaboration. This paper is focused on discussing levels of adoption of collaboration in these remote laboratory management systems.**

***Index Terms*—remote laboratories, e-learning, collaborative systems**

## I. INTRODUCTION

Remote laboratories enable the access to laboratories located in the host institution. If, as it is commonly the case, the equipment does not need human attention to work (e.g. as opposed to a robot laboratory where batteries are required, a laboratory with hazardous materials that requires supervision, etc.), then it can be run in a 24-hour basis. Students can access it during winter breaks, weekends, etc. at anytime from anywhere connected to the Internet. This increases the chances of decreasing the number of required equipment: in electronics, if 20 boards are required for 40 students when the boards are available few hours a week, 10 boards might be enough if they can access anytime.

A characteristic generally present in remote and traditional laboratories is that they are long time unused. In the case of remote laboratories, they can be easily shared with other institutions, regardless the city, country or continent where the student is. The sharing of remote laboratories through the federation of institutions makes sharing the costs of laboratories possible, either by paying for the access or by providing different laboratories. If two


**Figure 1 VISIR experiment running in WebLab-Deusto**

institutions each have a remote laboratory, they can exchange them so students would in fact have two laboratories for the same price. This way, federating remote laboratories would make it possible to increase the use made of the investment on laboratories, as well as pedagogically add more types of practices to the curricula of students.

In order to exploit such a federation of remote laboratories, a software infrastructure that makes it possible to develop, deploy, manage and share remote laboratories is required. Targeting this, different remote laboratory systems have arised: MIT iLabs[1] [1], Labshare Sahara[2] [2], WebLab-Deusto[3] [3], LiLa[4] [4] or VLCAP [6].

However, these software infrastructures typically focus on a set of students accessing a set of experiments. While some efforts have been placed in the scope of Virtual Worlds [7][8] (see Figure 2), it does not seem to have been included in the core of these infrastructures yet.

Indeed, some complex problems may arise. For instance, one of the advantages of these infrastructures is that they can balance the load of users among different copies of the same experiment. In the case of WebLab-Deusto, this is even possible in a federated way. When students select a particular laboratory, they might go to one copy or other. For certain laboratories, the collaboration should be established at laboratory level: two collaborating users sharing resources should be

[1] http://ilab.mit.edu/
[2] http://www.labshare.edu.au/
[3] http://www.weblab.deusto.es/
[4] http://www.lila-project.org

interacting with the same equipment and not with other copy. In other laboratories, students may not need to use the same particular copy, but only exchange specifications of what they are doing and see each one what is going on. The contribution of this paper is studying what problems will arise when dealing with collaboration from a software perspective.

**Figure 2 SecondLab: a SecondLife based Remote Lab**

## II. IMPLEMENTING COLLABORATIVE EXPERIMENTS IN WEBLAB-DEUSTO

WebLab-Deusto is a Remote Laboratory Management System (RLMS) which supports collaborative experiments. In this section, it will be detailed how this collaboration is implemented from a technical point of view.

### *A. Remote Laboratory Management Systems*

Many of the features developed in a remote laboratory (authentication, authorization, scheduling, user tracking...) are reusable from one remote laboratory to other. Due to this reason, information systems that provide these services have been created. As already mentioned, examples of these systems are: MIT iLabs, Labshare Sahara and WebLab-Deusto. These systems are independent of a particular setting, so they may be refered as general purpose remote laboratories in contrast to specific purpose remote laboratories. However, given that most of their work is adding management layers, these systems will be refered as Remote Laboratory Management Systems (RLMS). RLMSs come with guidelines and software development toolkits to develop remote experiments. This way, RLMSs speed up the development process of remote laboratories: teachers aiming to create a remote laboratory do not need to work on scheduling, authentication, authorization, etc. but focus on making the experiment available through the Internet. Additionally, RLMSs provide administration tools, so teachers can use them to add students, grant permissions on certain laboratories, track students, etc. The advantage of embracing these systems is that once they add a new transversal feature, all the experiments developed with that RLMS will automatically have it. For instance, if a RLMS did not support LDAP (a directory protocol common in universities which is used to authenticate and authorize users), and in the next release it supports it, automatically all the experiments developed with that RLMS will support it. Indeed, this flexibility of RLMSs is interesting enough to create efforts to integrate existing remote laboratories on RLMSs. For instance, in [4], the VISIR remote laboratory is integrated in MIT iLabs, or in [3] it is integrated in WebLab-Deusto.

### *B. WebLab-Deusto*

WebLab-Deusto is an open source Remote Laboratory Management System, developed in the University of Deusto, and built on top of open source technologies. It has been used by students as part of their courses since February 2005 through different versions. On top of it, teachers can develop remote laboratories, benefitting from all the features provided by the system, and then deploy them in their university. Furthermore, they can even share these laboratories with other WebLab-Deusto instances deployed in other universities or secondary schools.

A detailed description of the features of WebLab-Deusto has already been addressed in the literature [3]. They can be summarized in that it manages authentication (checking if the user is who claims to be), authorization (establishing who is granted to use which laboratories), scheduling (enqueuing the users among the copies of the laboratories available) user tracking (reporting who used what) and sharing (enabling other WebLab-Deusto instances to use certain laboratories).

As Figure 3 shows, WebLab-Deusto uses a layered architecture. Students connect to the core servers, which manage the transversal features, and these servers finally rely on a set of experiment servers that could be spread the campus. There are two approaches for developing laboratories in WebLab-Deusto: using managed experiments and using unmanaged experiments. In the former, the communication is managed by WebLab-Deusto, so it is secured and tracked by the system. However, it must be based on exchanging messages, and since this is not always possible (with protocols such as Remote Desktop, VNC or LabVIEW Remote Panels), WebLab-Deusto supports unmanaged experiments. On these experiments, the communication with the Experiment Server is direct, avoiding latency but decreasing the level of security and user tracking.

In the case of the managed experiments, WebLab-Deusto provides libraries for multiple programming platforms (Java, .NET, C++, C, Python, LabVIEW…) to develop experiment servers in WebLab-Deusto. This is so given the amount of technologies used in remote laboratories [9]. In order to secure the communication with these experiment servers, as well as to perform complex operations (such as checking that the experiments are still alive to report administrators of failures), a middle layer called "laboratory server" was developed. This laboratory server is provided by WebLab-Deusto and it proxies the functionality of a set of experiment servers, securing them.

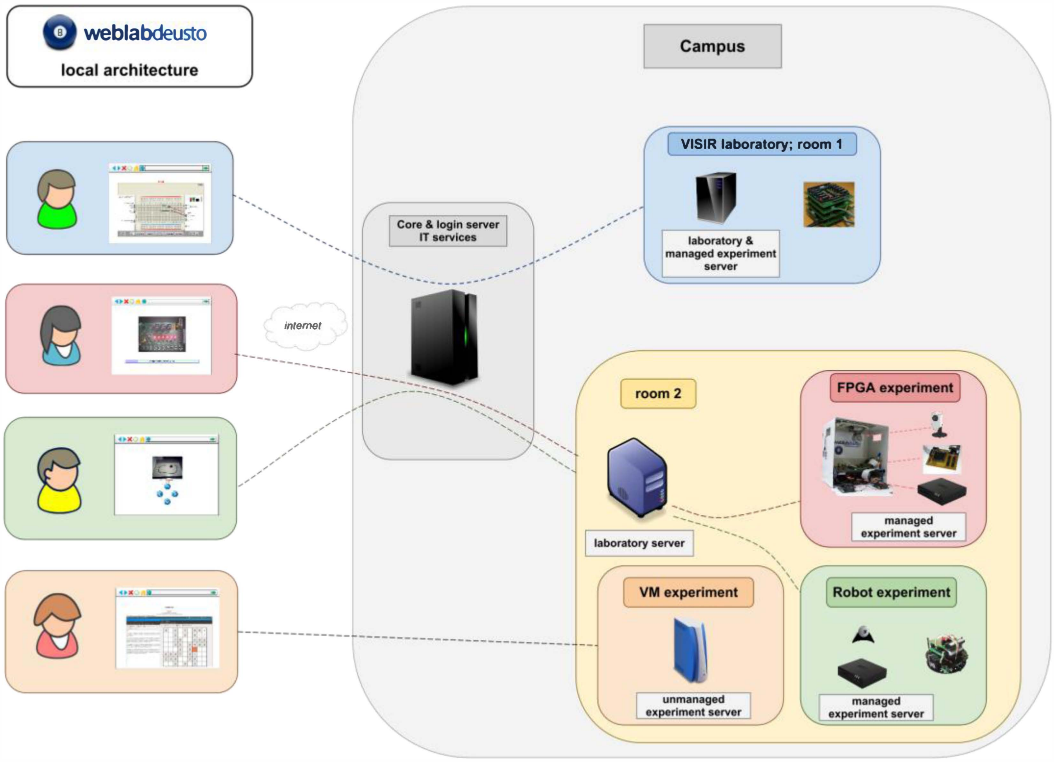


**Figure 3 Local architecture of WebLab-Deusto**

This way, the experiment development becomes simpler, since the experiment developer (normally a teacher) can choose the programming language and they know that no other student is accessing the same copy of the laboratory at the same time. The basic interface for communication (there are other methods for the scheduler to know the state of the equipment and other meta-information) at the Experiment Server is:

- start_experiment(client_data, server_data) => initial_configuration
- send_command(command) => response
- send_file(command) => response
- dispose() => response

While this process makes it easy to develop experiments, the fact of it being based on exclusive access made collaboration impossible.

### C. Implementing collaboration

In order to achieve collaboration, it was required to provide an additional interface which is a modified version of the interface presented above, aiming to let the experiment developer to effectively separate different concurrent students. So as to differentiate between them, a new argument is passed, providing a unique string per user session:

- start_experiment(session_id, client_data, server_data) => initial_configuration
- send_command(session_id, command) => response
- send_file(session_id, command) => response
- dispose(session_id) => response

The experiment developer will select which interface will use to implement the experiment, so it will be defined whether the experiment is collaborative or not. Then, the scheduler also provides new options to be able to establish how many concurrent students can enter in a single experiment.

This way, it becomes possible to develop traditional exclusive experiments as well as collaborative experiments. The following section will detail how this new interface can be used.

## III. Case Studies

This section presents two case studies of how this API can be used: the VISIR electronics remote laboratory, where the API is used and the Robotics control laboratory, which has not yet been adapted to the collaborative interface.

### A. Case study I: VISIR

VISIR [10] is an electronics remote laboratory (see Figure 1), deployed in 6 different universities in Europe. VISIR enables students to build electronic circuits in a protoboard, using different components such as resistors and capacitors, and link them to function generators, oscilloscopes, etc. VISIR provides a commutation matrix based on relays which make it possible that every time a student attempts to take a measurement, the system builds the desired circuit, takes the measurement, and returns the data to the user in a fraction of second. This way, multiple students can be working with different circuits at the same time, multiplexing the amount of users in time. However, a big amount of users will degrade the user experience, since students will wait for several seconds before retrieving the desired value. For this reason, a maximum number of concurrent users is used, which might range between 16 and 60 depending on the components established.

WebLab-Deusto has integrated VISIR as a regular experiment, first through the standard interface (proxying all the requests) and now through the collaborative interface. The advantage is that now it is possible to make different students interact each other by publishing the circuits they have built. As seen on Figure 4, a new button is added to "publish" a circuit. This circuit will automatically be available to other students connected, who will be able to load the circuit in their screen, by clicking on the "Circuits available" list.

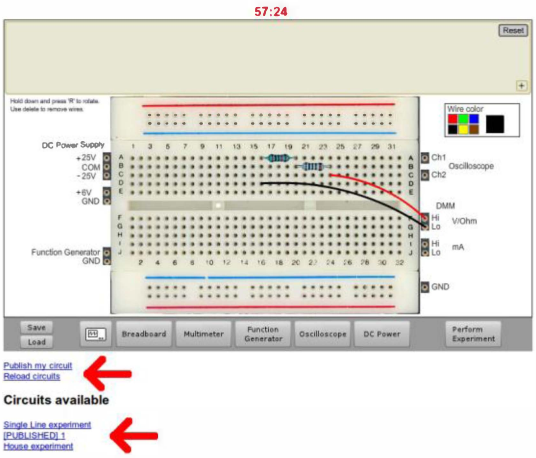


**Figure 4: VISIR used through WebLab-Deusto, with "Publish my circuit" option**

### B. *Case study II: Robotics control*

WebLab-Deusto nowadays counts with a robotics experiment, where students learn how to program a microbot like the one shown in Figure 2. The concept of "room" (as in chat systems) would fit in this experiment, where students would be accessing the same robot and using it at the same time. Concepts such as kicking the other users in the room would not make sense in the VISIR experiment, but could be present here. Additionally, inviting someone from the queue would also fit in.

## IV. CHALLENGES

An interesting point of VISIR is the fact that it has been replicated in different universities. Since WebLab-Deusto supports sharing remote laboratories through federations, if the VISIR at Deusto supports 60 concurrent users, and the VISIR at UNED supports 60 concurrent users, WebLab-Deusto at Deusto will talk to WebLab-Deusto at UNED and, if there is room available, it will dynamically transfer them some students if they were queued in Deusto, balancing the load of users among different federated copies.

The interest on this unique characteristic of remote laboratories -federating them to increase the types of practices and reduce costs- is growing. The Labshare project survey [11], made on all 34 Australian universities offering undergraduate engineering programs, reflects that interviewed executives were more interested in getting involved for the pedagogic merits of the remote laboratories, and that they were more inclined on initially being laboratory consumers than providers. Indeed, the European Union Commission is going to spend 60 million euro in research actions, projects and network of excellences in Technology enhanced Learning (TEL), under the objective ICT-2011.8.1 of the call FP7-ICT-2011-8. One of the target outcomes is precisely supporting a European-wide federation and use of remote laboratories and virtual experimentations for learning and teaching purposes.

However, given that the collaborative features developed on top of the VISIR integration are defined at Experiment Server level, students transferred to UNED will not be able to share their circuits with those users which are using the system at Deusto. This makes federation in general, and federated load balance in particular, a source of problems in terms of collaboration.

In the case of the robot, there could be several copies of the same robot, and students would be balanced among them through WebLab-Deusto. If two students are collaborating to see what the behavior of the robot is, they want to use the same particular copy. However, given that the collaborative interface works at Experiment Server level, the scheduler may not be aware of this and both students could be using two different robots while waiting for each other. This implies that in any case the scheduler must include some collaborative features in its very core to support this kind of scenario. Furthermore, the problems arisen from the federation model detailed in the previous case would become a problem in this scenario too.

## V. CONCLUSIONS AND FUTURE WORK

This contribution presents an initial approach for enabling collaboration in Remote Laboratory Management Systems (RLMS). Furthermore, it shows how it has implemented in one experiment (VISIR) and how it could be implemented in other (robotics control laboratory), outlining the challenges that arise from this approach.

The approach relies on enabling remote laboratory developers using WebLab-Deusto to distinguish among different concurrent users, so they can develop the logic for communicating among them. The problem of this approach is that, given that it is developed at experiment level, different copies of the laboratory may not be able to communicate, especially when they are working in a federated environment. This is a clear limitation of this approach, while it is useful for those experiments which are not going to be replicated.

Future work is divided in two independent ways. First, WebLab-Deusto will convert more experiments to the new collaborative API. The concept of shared rooms is especially interesting for certain experiments subject to be collaborative. Second, efforts to make it possible to communicate collaboration in federated environments (where different universities may have even different versions of the same system) will be considered.

## REFERENCES


[1] V. J. Harward, J. A. del Alamo, S. R. Lerman, P. H. Bailey, J. Carpenter, K. DeLong, C. Felknor, J. Hardison, B. Harrison, I. Jabbour et al.,"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, 2008.

[2] R. Sarukkalige, E. Lindsay, and A. Anwar, "Laboratory demonstrators perceptions of the remote laboratory implementation of a fluid mechanics laboratory," 2010.

[3] P. Orduna, J. Irurzun, L. Rodriguez-Gil, J. Garcia-Zubia, F. Gazzola, and D. Lopez-de Ipiña, "Adding new features to new and existing remote experiments through their integration in weblab-deusto," International Journal of Online Engineering (iJOE), vol. 7, no. S2, pp. pp–33, 2011.

[4] D. Zutin, M. Auer, "Work in progress—Integrating educational online lab platforms around the iLab Shared Architecture" in Frontiers in Education Conference (FIE), 2011 (IEEE, 2011).

[5] T. Richter, D. Boehringer, and S. Jeschke, "Lila: A european project on networked experiments," Automation, Communication and Cybernetics in Science and Engineering 2009/2010, pp. 307–317, 2011.

[6] R. Raman, P. Nedungadi, K. Achuthan, and S. Diwakar, "Integrating collaboration and accessibility for deploying virtual labs using vlcap," 2011.

[7] Scheucher, T. and Bailey, P.H. and Gütl, C. and Harward, V.J. Proceedings of the International Conference of Remote Engineering and Virtual Instrumention (REV2009). 2009.

[8] Garcia-Zubia, J. and Irurzun, J. and Angulo, I. and Hernández, U. and Castro, M. and Sancristobal, E. and Orduña, P. and Ruiz-de-Garibay, J. SecondLab: A remote laboratory under Second Life. Education Engineering (EDUCON), 2010 IEEE. 2010.

[9] C. Gravier, J. Fayolle, B. Bayard, M. Ates and J. Lardon, State of the Art About Remote Laboratories Paradigms - Foundations of Ongoing Mutations. International Journal of Online Engineering, Vol 4, No 1 (2008).

[10] Gustavsson, I. and Zackrisson, J. and Haakansson, L. and Claesson, I. and Lagö, T. The VISIR project--an open source software initiative for distributed online laboratories. Proceedings of the REV 2007 Conference, Porto, Portugal. 2007.

[11] T. Kostulski, S. Murray. "The National Engineering Laboratory Survey – A review of the delivery of practical laboratory education in Australian undergraduate engineering programs". December 2010. Part of the outcomes of the LabShare Project: http://www.labshare.edu.au/media/img/labshare_report_panel_website.pdf

## AUTHORS

**Pablo Orduña** is with the Deusto Institute of Technology - DeustoTech, University of Deusto, Avda. Universidades 24, 48007 Bilbao, Spain (e-mail: pablo.orduna@deusto.es).

**Javier Garcia-Zubia** is with the Faculty of Engineering, University of Deusto, Avda. Universidades 24, 48007 Bilbao, Spain (e-mail: zubia@deusto.es).

**Luis Rodriguez-Gil** is with the Faculty of Engineering, University of Deusto, Avda. Universidades 24, 48007 Bilbao, Spain (e-mail: luis.rodriguez@opendeusto.es).

**Ignacio Angulo** is with the Deusto Institute of Technology - DeustoTech, University of Deusto, Avda. Universidades 24, 48007 Bilbao, Spain (e-mail: ignacio.angulo@deusto.es).

**Olga Dziabenko** is with the Deusto Institute of Technology - DeustoTech, University of Deusto, Avda. Universidades 24, 48007 Bilbao, Spain (e-mail: olga.dziabenko@deusto.es).

**Diego López-de-Ipiña** is with the Deusto Institute of Technology - DeustoTech, University of Deusto, Avda. Universidades 24, 48007 Bilbao, Spain (e-mail: dipina@deusto.es).