Computers in Human Behavior 30 (2014) 389–395

Contents lists available at SciVerse ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Towards federated interoperable bridges for sharing educational remote laboratories

Pablo Orduña [a,*], Philip H. Bailey [b], Kimberly DeLong [b], Diego López-de-Ipiña [a,1], Javier García-Zubia [c]

[a] Deusto Institute of Technology – DeustoTech, University of Deusto, Bilbao, Spain
[b] Center for Educational Computing Initiatives, Massachusetts Institute of Technology, MA, USA
[c] Faculty of Engineering, University of Deusto, Bilbao, Spain

ARTICLE INFO

Article history:
Available online 31 May 2013

Keywords:
Remote laboratories
Federation
Interoperability

ABSTRACT

Educational remote laboratories are software and hardware tools that allow students to remotely access real equipment located in the university as if they were in a hands-on-lab session. Different initiatives have existed during the last two decades, and indeed toolkits (e.g. iLabs, WebLab-Deusto or Labshare Sahara) have been developed to ease their development by providing common management features (e.g. authentication or scheduling). Each of these systems was developed aiming particular constraints, so it could be difficult to migrate the labs built on top of one system to other. While there is certainly some overlap among these systems, with bridges among them they become complimentary. Given that these systems support web services based federation protocols for sharing labs, it is possible to achieve this goal, and share labs among different universities through different systems. The impact of this goal is that different institutions can increase the experiential activities of their students, potentially improving their learning goals. The focus is the integration of WebLab-Deusto labs inside the iLab Shared Architecture, as well as the integration of iLab batch labs inside WebLab-Deusto, detailing limitations and advantages of both integrations and showing particular cases.

1. Introduction

Remote laboratories enable the access to laboratories located in the host institution. These laboratories are physically located in the institution and comprehend many areas: physics (Gustavsson, Zackrisson, Håkansson, Claesson, & Lagö, 2007), chemistry (Coble et al., 2010), robotics (Dziabenko, García-Zubia, & Angulo, 2012), or even nuclear reactors (Hardison, DeLong, Bailey, & Harward, 2008).

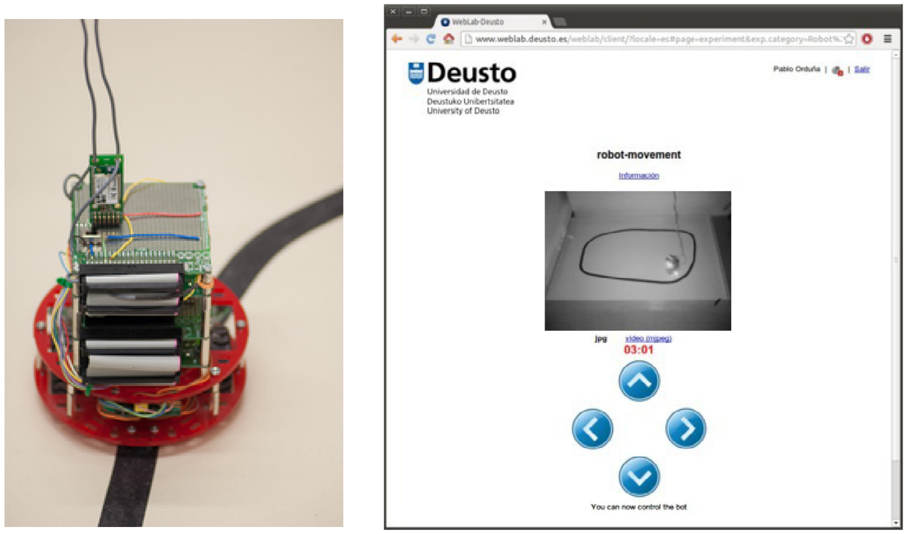
As a straightforward example, WebLab-Deusto provides a mobile robot shown in Fig. 1, which is controlled by a Microchip PIC microcontroller. Students learn how to program in PIC assembler, so the real robot is provided, as well as the instructions on what inputs and outputs are available. Students program the code, and while doing this, they can log in the WebLab-Deusto system through the Internet, submit the program to a real robot located in the University of Deusto. Then, for a small slot of time, they can exclusively see the effects of the program in the real robot. Other students attempting to use the laboratory will be queued until the current user finishes (in a matter of minutes). Finally, instructors can later check the usage of the laboratory, which programs have been sent, and gather statistics.

While remote laboratories cannot be used in all type of experiential learning (for instance, the laboratory would not be suitable for a lesson where students must learn how to build a robot), in those fields where it is suitable (e.g. learning how to program in assembler), it adds flexibility, since students can learn experimenting at night or during weekends without being in the university. Effectiveness of this type of laboratory has been already addressed in the literature (Lang et al., 2007; Corter, Esche, Chassapis, Ma, & Nickerson, 2011; Garcia-Zubia et al., 2011), which is outside the focus of this contribution. The focus of this contribution is on how to maximize the type of laboratories available for a particular institution, increasing the available experiential learning among its students. The effectiveness of this learning will depend on the particular laboratories shared among institutions.

So as to develop a remote laboratory, certain features can be shared with other remote laboratories. From the example presented, students (a) log in the system – authentication, (b) use a scheduling mechanism – a queue – to guarantee exclusive access, (c) communicate with the remote system, (d) do something particular of the laboratory – send the program, see the results – and (e) enable user tracking by the instructor. Except for the particular code, the rest of the features could be shared with a different type

* Corresponding author. Tel.: +34 94 413 9003x2977.
E-mail addresses: pablo.orduna@deusto.es (P. Orduña), pbailey@mit.edu (P.H. Bailey), kirky@mit.edu (K. DeLong), dipina@deusto.es (D. López-de-Ipiña), zubia@deusto.es (J. García-Zubia).
[1] Tel.: +34 94 413 9003x2977.

0747-5632/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.chb.2013.04.029



(a) Azkar robot (b) Robot controlled through a web interface

Fig. 1. Response times measured in different embedded devices.

of remote laboratory. So as to do this, remote laboratory management systems (RLMSs) were developed. They provide APIs (Application Programming Interfaces) to develop new remote laboratories, authentication and authorization mechanisms, scheduling systems, management tools (to add/remove users, laboratories, etc.), and user tracking tools. A laboratory developer can use these systems to create new laboratories easily, since the RLMS already provides these features. Additionally, upgrades of these RLMSs will provide the laboratory developer with more features.

Additionally, the creators of these laboratories found that, once students of a particular institutions can access through the Internet to a particular laboratory, it can also be accessed by students of other universities. This way, these systems supported federation. This means that these systems enable that two independent deployments of the management system in two institutions can share their laboratories automatically, under a service level agreement (SLA), such as "students of this university can access 10,000 a year this laboratory".

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 (Kotulski & Murray, 2010), 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 were more inclined on initially being laboratory consumers than providers. Indeed, the European Union Commission is investing 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.

Three major RLMSs can be found in the literature: MIT iLabs[2] (Harward et al., 2008b), WebLab-Deusto[3] (Orduña et al., 2011) and Labshare Sahara[4] (Diponio, Lowe, & de la Villefromoy, 2012). However, while these systems share the motivation, rationale and are essentially equivalent, technically each of them has been focused on a type of laboratory and have certain differences. For example, WebLab-Deusto has always been used with short session laboratories (i.e. students access often but in 3–10 min sessions), and therefore its main scheduling system is queueing, while MIT iLabs in its interactive version will rely on calendar-based booking for supporting long session laboratories. This is common given the wide background differences in remote laboratories in terms of technologies (Gravier, Fayolle, Bayard, Ates, & Lardon, 2008) and approaches to create the laboratories. In order to build an ecology of remote laboratories (Harward et al., 2008a), not only a software infrastructure is required, but also a deep understanding of the student audiences. Since each system has been influenced by different student audiences, building bridges between two systems, when feasible, make it possible for each system to consume laboratories designed for other audience.

In this line, (Yeung, Lowe, & Murray, 2010) proposed the LabConnector application protocol interface (API) as a bridge between iLabs and Labshare Sahara focused at protocol level, evaluating it with an iLab laboratory located in the University of Queensland being consumed by Labshare Sahara. While the bridge itself might have technical difficulties in becoming adopted by other systems, it represented a clear step forward in the interoperability of remote laboratory management systems.

In this contribution, a bridge to consume WebLab-Deusto laboratories by the iLab Shared Architecture (Harward et al., 2008b), as well as an experimental bridge to consume iLab batch laboratories from WebLab-Deusto is presented. This type of bridge make it possible that institutions can wide the number of supported remote laboratories, increasing the student audiences and supporting more laboratories. In order to achieve a global solution, an interface defined by the Global Online Laboratory Consortium (GOLC[5]) would be required.

2. Remote laboratory management systems

This section provides a brief summary of the architectures of MIT iLabs and WebLab-Deusto, focusing only on the most relevant parts for the contribution. Other remote laboratory management systems are outside the scope of this contribution.

[2] http://ilab.mit.edu.

[3] http://www.weblab.deusto.es.

[4] http://www.lila-project.org/.

[5] http://www.online-lab.org/.

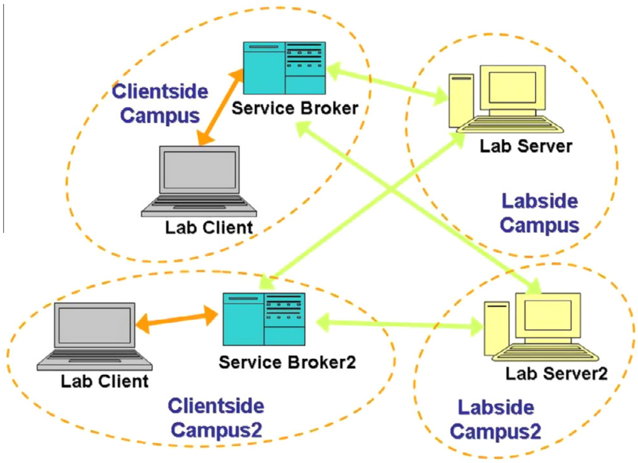


Fig. 2. Topology of the iLab batch laboratories.

2.1. MIT iLabs

The iLabs project (Harward et al., 2008b) is a remote laboratory management system developed in the MIT during over a decade, and used and deployed in the five continents. It was the first remote laboratory system in introducing the concept of laboratory federation: it enables the sharing of laboratory at architecture level, dividing the architecture in elements present in student-side campus and elements present in laboratory-side campus. It is based on a brokered architecture called iLab Shared Architecture (ISA). On it, each institution has a Service Broker. Local students are authenticated and identified as valid members of certain groups in the Service Broker of their institution, and the authorization to use laboratories relies on the permissions of the groups. Conceptually, students interact with servers in their institution, and these will interact with remote servers in federated institutions, unless a direct connection among the student with the final laboratory is required.

The ISA supports two types of laboratories (Hardison et al., 2008): batch laboratories and interactive laboratories, which have different topologies and scheduling mechanisms.

The batch laboratories are based on queues: students download a Lab Client that submits requests to the Service Broker of their institution, and this Service Broker will forward them if authorized to the Lab Server of the remote institution (see Fig. 2). The queuing is managed at this server, so the student will be polling through these servers to retrieve status information of the experiment submitted. In order to avoid students waiting in the queue for an invalid experiment, prior to submitting the experiment, the Lab Client may request a validation of the experiment by the server. The Lab Server will be able to reject it if it can without using the hardware. Finally, the Lab Server notifies the student's Service Broker that it has finished each experiment, so the Service Broker can retrieve and store the results.

The interactive laboratories are laboratories where students have direct access to the Lab Server during an amount of time. Given that this time tends to be long, they are based on booking instead of queueing. This way, students can use a calendar to book in advance a session, and when the system confirms it, they know that they will have exclusive access to the equipment during that session. The process of booking this session involves several servers in both the student side campus and the laboratory side campus (see Fig. 3).

Finally, the ISA supports that certain laboratories require no scheduling scheme to the ISA infrastructure, not being classified as batch nor interactive. This model is interesting for virtual laboratories instead of remote laboratories – where students do not access real hardware equipment- that can be accessed by several students at the same time.

From the ISA perspective, each laboratory is unique and therefore it does not support load balancing among different copies of the laboratory at infrastructure level. Depending on the type of laboratory, it can be developed at laboratory level (e.g. a batch Lab Server could implement it since it handles the queue).

2.2. WebLab-Deusto

WebLab-Deusto is a remote laboratory management system developed in the University of Deusto. In the first two versions it was an ad hoc remote laboratory (García-Zubía, López-de Ipiña, & Orduña, 2005) which became a remote laboratory system on top of which different laboratories could be built using different soft-

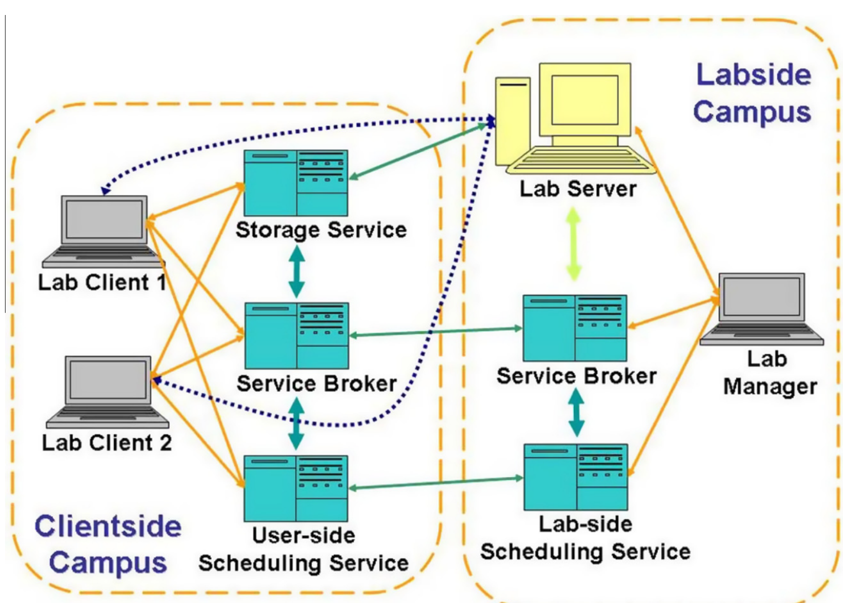


Fig. 3. Topology of the iLab interactive laboratories.

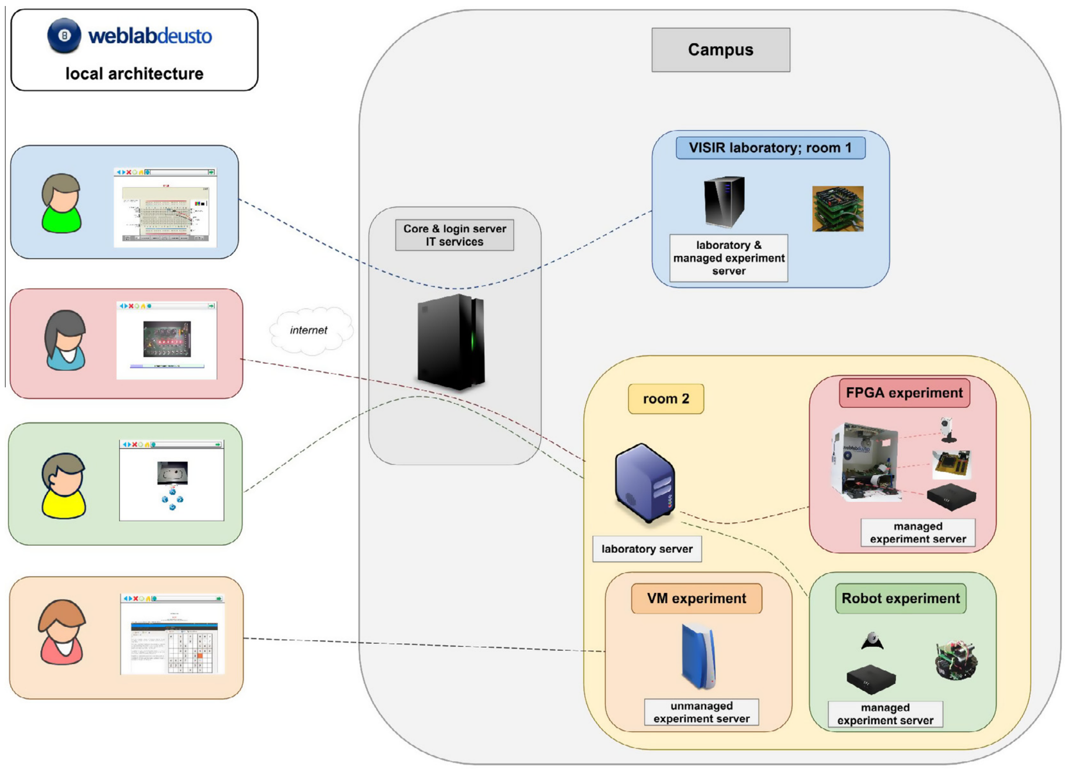


Fig. 4. General overview of the WebLab-Deusto.

ware technologies (Orduña et al., 2009) and nowadays supports federation of remote laboratories (Orduña, Rodriguez-Gil, Lopez-de Ipiña, & Garcia-Zubia, 2012).

The WebLab-Deusto architecture (see Fig. 4) is a client–server architecture. Students usually interact with the Login Server for authentication and with Core Server for the rest of the operations. The Core Server will handle the authorization, scheduling and storage of the information sent by students, as well as forwarding all the requests to the proper laboratory. Behind the Core Server, the WebLab-Deusto architecture defines two more servers: the Laboratory Server and the Experiment Server. The first one aims to be located in the physical laboratory room where multiple Experiment Servers are expected. Its purpose is mainly to grant the security of the Experiment Servers, as well as periodically check if they are still working, and act as a wrapper. The Experiment Server is the server which physically interacts with the hardware. Libraries to build Experiment Servers are provided for different platforms (C++, C, Java, Python, LabVIEW). All the interaction between client and servers is sent through web services. Either SOAP, JSON over HTTP and XML-RPC can be used.

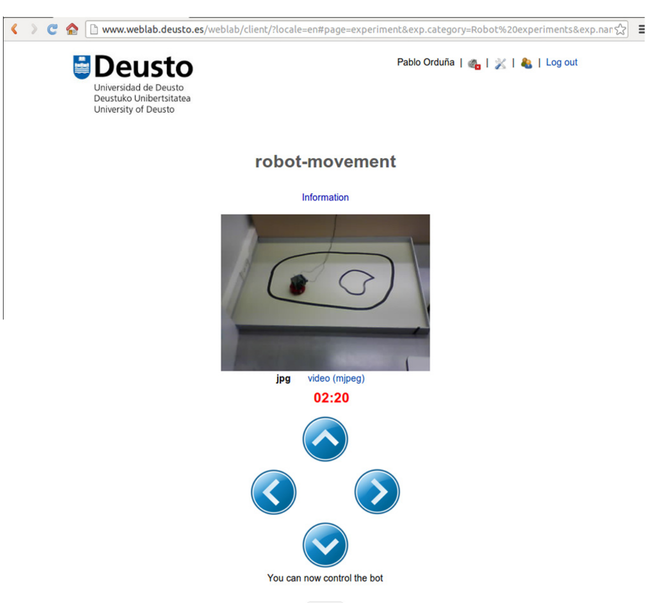


Fig. 5. WebLab-Deusto robot laboratory running through an iLab system.

Two types of laboratories can be built using WebLab-Deusto: managed and unmanaged laboratories. In the managed laboratories all the interaction between the student and the laboratory is managed through all the described layers. Therefore, the storage is done at the Core Server, and the Experiment Server does not need to be publicly available at network level. However, a command based protocol must be implemented in each laboratory. In the unmanaged laboratories, existing Experiment Servers manage other platforms (Orduña et al., 2011) such as Virtual Machines or LabVIEW Remote Panels. Therefore, students are still interacting with all the layers, but the main interaction is handled by the external platform (e.g. the Virtual Machine with VNC or Remote Desktop), and it is not stored or securized by WebLab-Deusto.

The scheduling system used is based on plug-ins. The main plug-in used is a priority queues system. It accepts different copies of the same laboratory, balancing the load of users among the different copies, so queues can be reduced by adding more copies. In order to federate different instances of WebLab-Deusto, a new scheduling plug-in was built, which basically relies on a set of external WebLab-Deusto instances. For instance, if three institutions have each two copies of the same laboratory, they can share

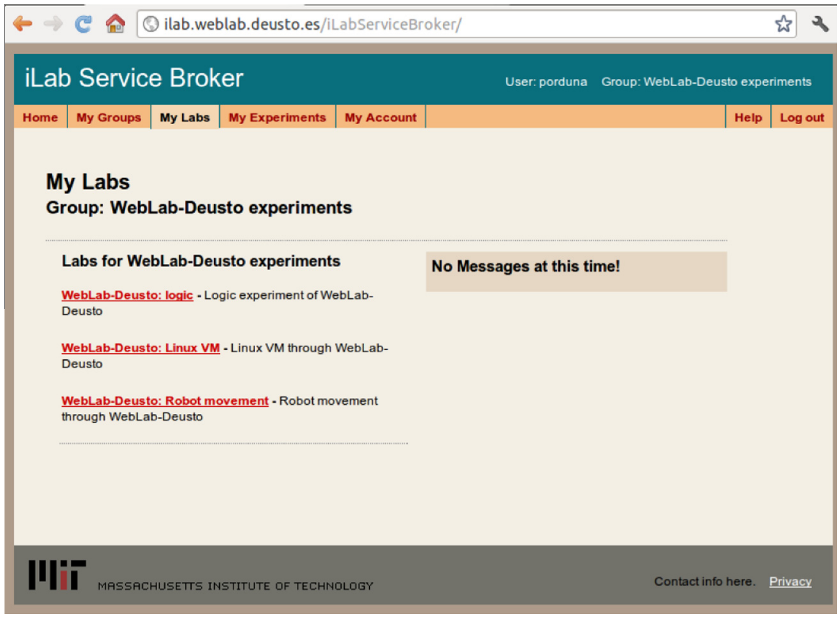


Fig. 6. WebLab-Deusto laboratories listed in iLab.

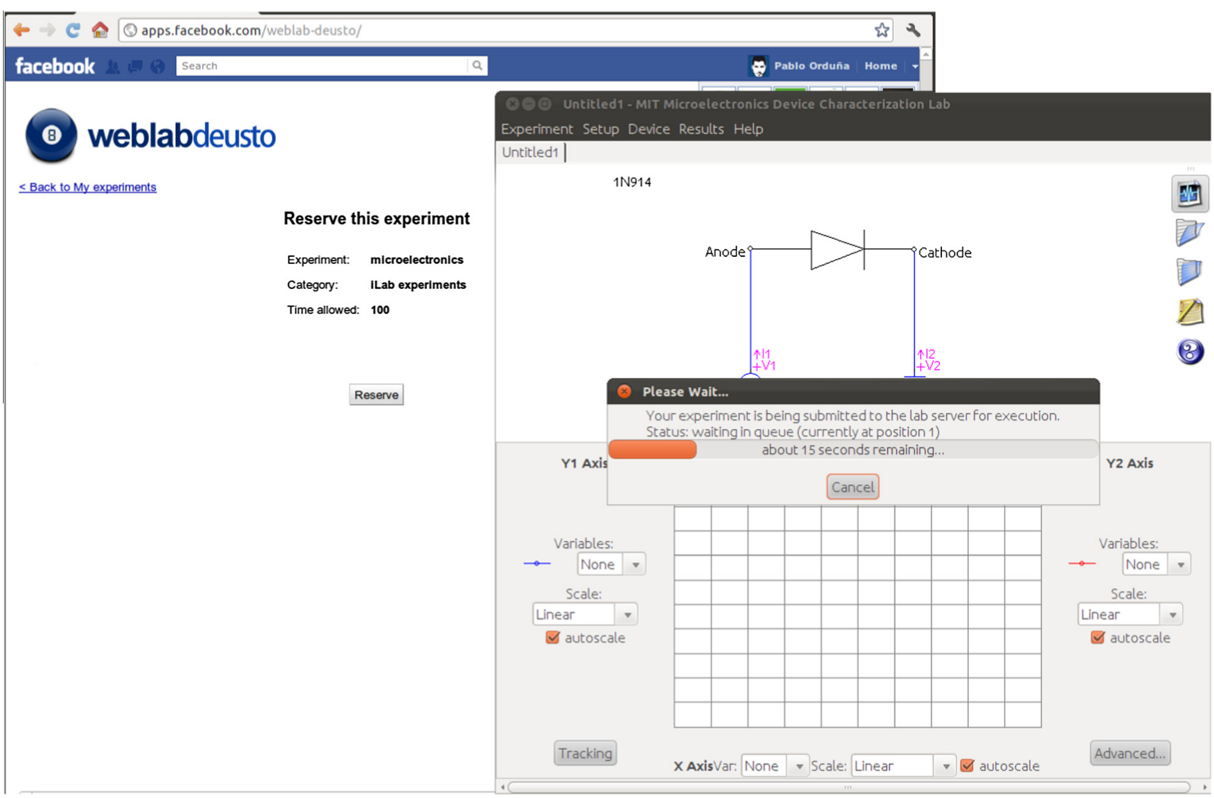


Fig. 7. MIT Microelectronics WebLab running through WebLab-Deusto in Facebook.

them among themselves with WebLab-Deusto. Each institution will be able to use their two resources first, but, if all of them are filled, they will rely on the other two institutions at the same time, balancing among both at the same time. At the same time, the institution can establish a higher priority to local students. Given the pluggable nature of the scheduling system, new developed plug-ins deployed in one of the institutions will automatically be available for federated students in the rest of the institutions. As the federation management plug-in is also a plug-in, three institutions could build a sharing chain: the first institution sharing laboratories with the second and the second institution re-sharing the same laboratories with the third one.

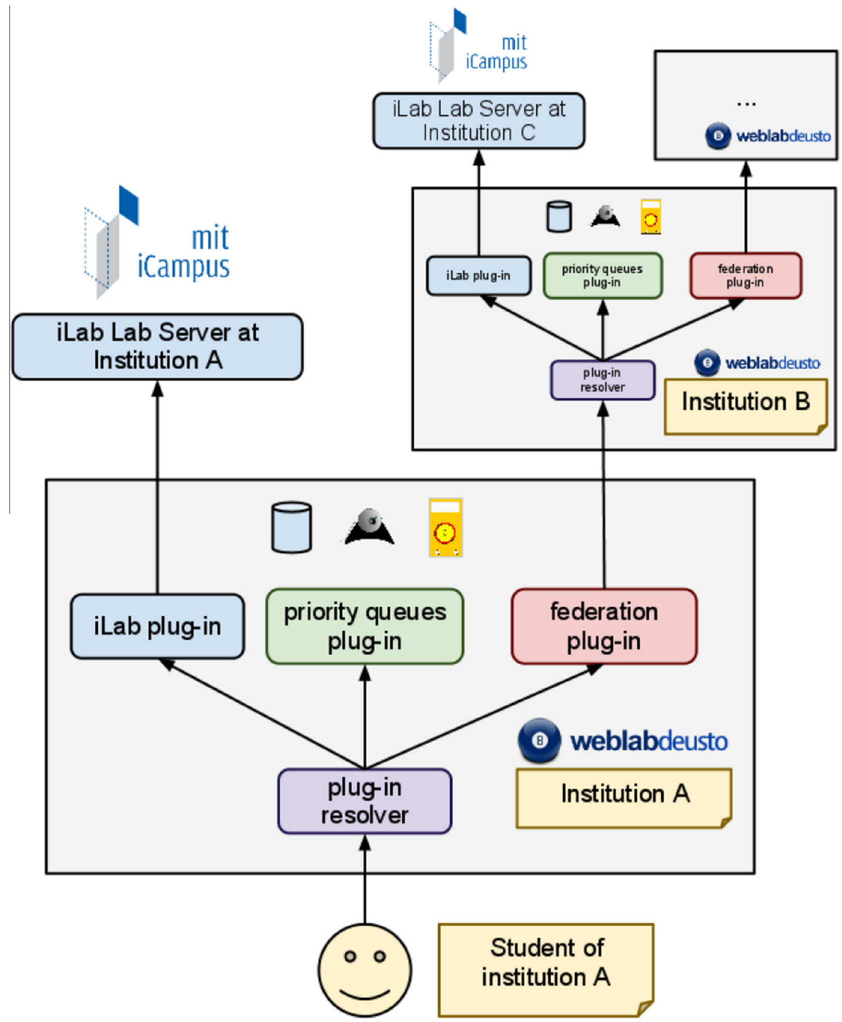


Fig. 8. Architectural overview of the iLab on WebLab-Deusto integration.

3. Bridging WebLab-Deusto and MIT iLabs

This section describes the bridges built to consume each others laboratories.

3.1. WebLab-Deusto laboratories in MIT iLabs

WebLab-Deusto supports batch and interactive (managed and unmanaged) laboratories, but all the laboratories used by students nowadays are interactive. Therefore, the target is to use WebLab-Deusto interactive laboratories in iLabs, regardless if they are managed or not. However, the ISA only supports scheduling through booking in the case of interactive laboratories, while WebLab-Deusto nowadays only supports queuing.

In order to handle this issue, the no scheduling option was selected in the ISA for the integration. Using this option, the ISA relies completely on the Lab Server, sending all the users that attempt to use the laboratory to the Lab Server. With this scheme, it was possible to develop a special type of Lab Server, which relied on WebLab-Deusto through web services. Therefore, every time a student tries to use the laboratory, the Lab Server will interact with WebLab-Deusto. In the configuration file of each Lab Server it is possible to define not only the credentials required and the desired laboratory type, but also advanced arguments such as the priority required or the time assigned in the remote system.

For instance, an iLab system could have three copies of this "glue" Lab Server, two of them configured to use a robotics laboratory, and another to use a FPGA laboratory. At ISA level, they are three different laboratories, even if they all use the same WebLab-Deusto system that manages three real laboratories. The two robotics laboratories could even be the same at WebLab-Deusto but splitted at ISA to manage different priorities. This way, if students of 2nd grade are assigned to a robotics laboratory and students of 4th grade are assigned to the other, the iLab administrator can choose which group has a higher priority over time, even if this priority is managed in the queues in the remote WebLab-Deusto system.

Being WebLab-Deusto laboratories presented as regular iLab laboratories, the authentication and authorization can be managed through the iLab tools (see Figs. 5 and 6). Furthermore, if the particular host institution desires to share the laboratories with other universities, it is possible using the ISA.

3.2. MIT batch iLabs laboratories in WebLab-Deusto

An experimental bridge of batch iLab laboratories into WebLab-Deusto has been implemented. This way, it becomes possible to consume iLab laboratories from WebLab-Deusto instances. So as to process the requests sent by iLab Lab Clients, a translator of a subset of the possible requests to their corresponding request type in WebLab-Deusto was developed. Once they have become WebLab-Deusto requests, WebLab-Deusto uses its pluggable scheduling system to handle the requests with a new scheduling plug-in for the iLab integration. This plug-in will convert the requests again and forward them to an external iLab Lab Server, therefore acting WebLab-Deusto as a Service Broker.

Given that it has been implemented as a plug-in in the core of the scheduling system of WebLab-Deusto, all the messages are automatically stored so educators can track the usage performed by students. The authentication and authorization is managed by WebLab-Deusto, so it can rely on systems supported by WebLab-Deusto. For instance, WebLab-Deusto supports creating accounts and being authenticated through OAuth 2.0 with Facebook, so as seen in Fig. 7, a student with permissions to use an iLab laboratory can run it from facebook.

Finally, being implemented as a plug-in also allows WebLab-Deusto to use other plug-ins of WebLab-Deusto, so inter-institutional chains can be built. As described in Fig. 8, students can access WebLab-Deusto in their institution (Institution A in the example), and through the plug-in resolver they can use an iLab plug-in contacting an iLab Lab Server in the same institution. Furthermore, through the federation plug-in, it is possible to connect WebLab-Deusto with other WebLab-Deusto in the Institution B, which has also set up the iLab plug-in with yet another institution (Institution C). More complex chains, even supporting distributed load balance, could be built with this approach.

However, the system is experimental since only a subset of request types have been implemented. In particular, only those request types required to validate, submit, wait and retrieve the results. Those request types required to store user information in their session have not been implemented. For instance, in the iLab Shared Architecture it is possible that a student stores the results retrieved for a future use. Since WebLab-Deusto lacks of this interesting feature, it could not be implemented in the bridge.

4. Conclusions and future work

The contribution has presented two bridges between WebLab-Deusto and the iLab Shared Architecture. While this work might not be directly extrapolated to other systems, it draws the potential advantages, drawbacks and the motivation for exploring this area. These bridges show that these TEL solutions can automatically federate laboratories that might not fit in one of the two systems or that would require a notable amount of work.

In both bridges, the major advantage is that students already used to one system can consume laboratories of the other using the solution they know and for which they already have credentials. Students and educators of one institution who are using one system with the laboratories developed on it for different classes can start using laboratories developed in the other, which will appear in the same menus.

In the case of WebLab-Deusto being consumed by the iLab Shared Architecture, it enables the use of WebLab-Deusto using the iLab federation model. This means that if an iLab Service Broker has a number of WebLab-Deusto laboratories included, it can share these to other iLab Service Broker. For instance, the Service Broker at the University of Deusto can share WebLab-Deusto laboratories through the iLab system to the iLab-Europe Service Broker,[6] so users there will automatically be able to consume those laboratories. Additionally, the iLab Shared Architecture benefits from three features provided by WebLab-Deusto: (a) consuming other federation model, so users of an iLab Service Broker bound to a WebLab-Deusto instance will be able to use laboratories of other WebLab-Deusto instance if the two WebLab-Deusto instances are federated; (b) support of queue based interactive laboratories; and (c) load balancing of laboratories among different copies – i.e. if there are two copies of one WebLab-Deusto laboratory, it will manage the queues so the load of users will be balanced in a transparent way for the iLab system-.

Regarding future work, the WebLab-Deusto laboratories consumed by the iLab Shared Architecture does not store the results in the iLab system, so educators can see that a particular student used the system but not what was done. The iLab Shared Architecture consumed by WebLab-Deusto also presents problems, since the interactive version – which requires booking – is not supported, as well as some features of the batch version such as storing information of each student. This is mainly due to limitations of the WebLab-Deusto system.

Acknowledgement

This work has been supported by research grants IPT-2011-1558-430000 (mCLOUD), funded by the Spanish Ministerio de Ciencia e Innovación and IE11-316 (Future Internet II), funded by the Basque Government.

References

Coble, A., Smallbone, A., Bhave, A., Watson, R., Braumann, A., & Kraft, M. (2010). Delivering authentic experiences for engineering students and professionals through e-labs. In *Education engineering (EDUCON)* (pp. 1085–1090). IEEE.

Corter, J., Esche, S., Chassapis, C., Ma, J., & Nickerson, J. (2011). Process and learning outcomes from remotely-operated, simulated, and hands-on student laboratories. *Computers & Education, 57*, 2054–2067.

Diponio, M., Lowe, D., & de la Villefromoy, M. (2012). *Supporting local access to collections of distributed remote laboratories*.

Dziabenko, O., García-Zubia, J., & Angulo, I. (2012). Time to play with a microcontroller managed mobile bot. In *Global engineering education conference (EDUCON)* (pp. 1–5). Springer.

García-Zubía, J., López-de Ipiña, D., & Orduña, P. (2005). Towards a canonical software architecture for multi-device weblabs. In *Industrial electronics society, 2005. IECON 2005. 31st Annual conference of IEEE* (pp. 6). IEEE.

Garcia-Zubia, J., Orduna, P., Angulo, I., Hernandez, U., Dziabenko, O., Lopez-Ipina, D., et al. (2011). Application and user perceptions of using the Weblab-Deusto-PLD in technical education. In *Frontiers in education conference (FIE)*. IEEE, pp. GOLC1–1.

Gravier, C., Fayolle, J., Bayard, B., Ates, M., & Lardon, J. (2008). State of the art about remote laboratories paradigms-foundations of ongoing mutations. *iJOE, 4*.

Gustavsson, I., Zackrisson, J., Håkansson, L., Claesson, I., & Lagö, T. (2007). The visir project – an open source software initiative for distributed online laboratories. In *Proceedings of the REV 2007 conference*, Porto, Portugal.

Hardison, J., DeLong, K., Bailey, P., & Harward, V. (2008). Deploying interactive remote labs using the iLAB shared architecture. In *Frontiers in education conference, 2008. FIE 2008. 38th Annual* (pp. S2A-1). IEEE.

Harward, V. J., del Alamo, J. A., Lerman, S. R., Bailey, P. H., Carpenter, J., DeLong, K., et al. (2008b). The iLAB shared architecture: A web services infrastructure to build communities of internet accessible laboratories. *Proceedings of the IEEE, 96*, 931–950.

Harward, V.J., del Alamo, J.A., Ayodele, K., Bailey, P.H., DeLong, K., Hardison, J., & et al. (2008a). Building an ecology of online labs. In *Proceeding of the international conference on interactive collaborative learning – ICL2008*.

Kotulski, T., & Murray, S. (2010). *The national engineering laboratory survey*. Labshare Project. December.

Lang, D., Mengelkamp, C., Jäger, R., Geoffroy, D., Billaud, M., & Zimmer, T. (2007). Pedagogical evaluation of remote laboratories in emerge project. *European Journal of Engineering Education, 32*, 57–72.

Orduña, P., Rodriguez-Gil, L., Lopez-de Ipiña, D., & Garcia-Zubia, J. (2012). Sharing the remote laboratories among different institutions: A practical case. In: *2012 9th International conference on remote engineering and virtual instrumentation (REV)* (pp. 1–4).

Orduña, P., García-Zubia, J., Irurzun, J., Sancristobal, E., Martın, S., Castro, M., et al. (2009). Designing experiment agnostic remote laboratories. *Remote Engineering and Virtual Instrumentation*.

Orduña, P., Irurzun, J., Rodriguez-Gil, L., Garcia-Zubia, J., Gazzola, F., & López-de Ipiña, D. (2011). Adding new features to new and existing remote experiments through their integration in Weblab-Deusto. *International Journal of Online Engineering (iJOE), 7*, p. 33.

Yeung, H., Lowe, D., & Murray, S. (2010). Interoperability of remote laboratories systems. *International Journal of Online Engineering (iJOE), 6*, 71.

[6] http://www.ilab-europe.net/.