Generic integration of remote laboratories in learning and content management systems through federation protocols

Pablo Orduña*, Sergio Botero Uribe†, Nicolas Hock Isaza‡, Elio Sancristobal§,
Mikel Emaldi*, Alberto Pesquera Martin§, Kimberley DeLong‡, Philip Bailey‡,
Diego López-de-Ipiña*, Manuel Castro§ and Javier Garcia-Zubia*
*DeustoTech - Deusto Institute of Technology
University of Deusto, Bilbao, Spain
Email: {pablo.orduna,m.emaldi,dipina}@deusto.es
† EAFIT, Medellin, Colombia
Email: sbotero2@eafit.edu.co
‡Center for Educational Computer Initiatives
Massachusetts Institute of Technology, Cambridge, MA. USA.
Email: {nhocki,kirky,pbailey}@mit.edu
§Electrical and Computer Engineering Department (DIEEC)
Spanish University for Distance Education (UNED), Madrid, Spain
Email: {elio,mcastro}@ieec.uned.es, apm@innova.uned.es

***Abstract*—Educational remote laboratories are a software and hardware tool that allows students to remotely access real equipment located in universities as if they were in a hands-on-lab session. Their integration in Content and Learning Management Systems (CMSs or LMSs) has been an active research topic for years, supporting mainly ad hoc solutions. A notable exception has been the use of federation protocols –commonly used for sharing laboratories from one university to other–, for actually sharing laboratories from a remote laboratory system to a C/LMS. This approach opened new doors in the simplification of the process, since it did not require the remote laboratories to make any type of change. The focus of this contribution is to provide a solution to decrease the number of functionalities required for creating an integration by providing a software component that reuses them. As shown in the contribution, this component has been implemented and two remote laboratory management systems (which provide access to multiple remote laboratories) are already supported, and a third one is under development. In the C/LMS side, all the LMSs supporting IMS LTI are supported, and HTTP APIs are provided for being supported by other systems. Indeed, the contribution describes its support in the Joomla CMS and in the Moodle 1.9 and dotLRN LMSs which do not support IMS LTI. The solution, called gateway4labs, is an open source initiative which targets to be used in production.**

I. Introduction

An Educational Remote Laboratory is a software and hardware solution that enables students to access real equipment located in their institution, as if they were in a hands-on-lab session. There are many kinds of remote laboratories, including fields such as Physics, Electronics, Robotics or Chemistry. Once these laboratories are available through the Internet, it becomes possible to share them with other universities. Given that both traditional and remote laboratories pass long time unused (nights, weekends, weeks which no practical lessons…), if *University A* shares two laboratories with *University B* and vice versa, then both have 4 labs with similar maintenance costs. For this reason, remote laboratories federations have been explored and been used in production for years.

Integrations of remote laboratories on Content Management Systems (CMSs) or Learning Management Systems (LMSs) –in the rest of the document, C/LMS– have been addressed in the literature. The focus is to integrate remote laboratories (which are educational resources after all) into LMSs (which is where the students and teachers meet and where students find other educational resources). While this integration has always been a relevant topic in the area of remote laboratories, most solutions in the literature have been ad-hoc: this remote laboratory developed in *University A* is integrated in the LMS used in University A and will not work with other C/LMSs. Not supporting this integration forces remote laboratory systems to duplicate certain structures (e.g., authorization) commonly already available in the C/LMS. Additionally, the user experience would be improved if the remote laboratories were treated as yet another educational resource in the LMS.

Last year, the usage of remote laboratories federations for the integration of remote laboratories in C/LMSs was proposed and evaluated [1]. In that contribution, a particular remote laboratory management system (WebLab-Deusto) was integrated in one LMS (Moodle) and one CMS (Joomla) to validate that federation protocols enable this integration in a simple approach. While the approach of using federation for this integration was novel, and it was noticeable that it did not require changes on the remote laboratory management system, it still required much of the logic to be implemented as plugins in the different C/LMSs.

The focus of this contribution is to present a novel and open

978-1-4673-5261-1/13/$31.00 ©2013 IEEE

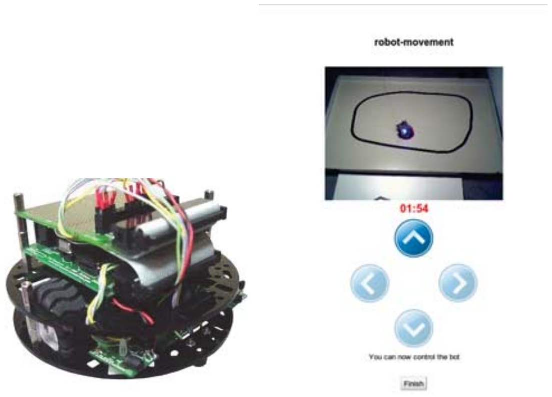


Fig. 1. Robot laboratory [2]. At the left, the mobile robot itself. At the right, the user interface once the program has been submitted.

source software component, called gateway4labs, that makes it possible to support different LMSs through the use of IMS LTI, as well as other systems (such as LMSs not supporting LTI, or CMSs) with lighter plug-ins than it was in that contribution. This component also supports a set of plug-ins for integrating new remote laboratory management systems, so it is extensible in both sides. It has been evaluated testing it in Moodle LMS (through IMS LTI), the .LRN LMS (without IMS LTI) and the Joomla CMS.

The paper is structured as follows: Section II introduces the concepts of remote laboratory, remote laboratory management system and remote laboratory federation. Then, Section III shows what integrations of remote laboratories in C/LMSs are available in the literature. Section IV explains the proposed solution and the analyzed use cases. Finally, Section VI describes the conclusions and the future work.

II. Remote Laboratories Federations

This section introduces the concepts of Remote laboratories, Remote Laboratory Management Systems and remote laboratory federations.

A. Remote Laboratories

A remote laboratory is a software and hardware tool that allows students to remotely access real equipment located in the university. Users access this equipment as if they were in a traditional hands-on-lab session, but through the Internet. To show a clear example, Figure 1 shows a mobile low cost robot laboratory described in [2]. Students learn to program a Microchip PIC microcontroller, and they write the code at home, compile it with the proper tools, and then submit the binary file to a real robot through the Internet. Then, students can see how the robot performs with their program through the Internet (e.g., if it follows the black line according to the submitted program, etc.) in a real environment.

In this line, there are many examples and classifications in the literature [3], [4]. Indeed, remote laboratories were born nearly two decades ago [5], [6], [7], and since then they have been adopted in multiple fields: chemistry [8], [9], physics [10], [11], electronics [12], [13], robotics [14], [15] and even nuclear reactor [16].

B. Remote Laboratory Management Systems

Every remote laboratory manages at least a subset of the following features: authentication, authorization, scheduling users to ensure exclusive accesses -typically through a queue or calendar-based booking-, user tracking and administration tools. These features are common to most remote laboratories, and are actually independent of the particular remote laboratory settings. For example, an authentication and queuing system is valid both for an electronics laboratory and for a chemistry laboratory.

For this reason, Remote Laboratory Management Systems (RLMSs) arose. These systems (e.g., MIT iLabs[1], WebLab-Deusto[2] or Labshare Sahara[3]) provide development toolkits for developing new remote laboratories, as well as management tools and common services (authentication, authorization, scheduling mechanisms). The main idea is that by adding a new feature to one of them (e.g., supporting LDAP, or LMSs), all the laboratories which are developed on top of them will support this feature automatically.

C. Federating Remote Laboratories

One of the features that RLMSs started supporting was federating their remote laboratories. For example, if two universities (*University A* and *University B*) install a particular RLMS, they support federation protocols so *University A* shares a laboratory with students of *University B* without knowing these students. The key here is that the provider university does not need to register particular students, but rather groups or simply universities. It is the consumer system who defines that a set of local users can access a particular laboratory of the provider system.

Therefore, the relationship between two federated entities is the following:

- The consumer system manages the authentication and authorization of its students.
- The provider system manages the scheduling and the access to the laboratories, storing what the users did.
- The consumer system will later ask for results to the provider system.
- In every moment, the provider system does not need to know anything related to the particular students.

D. Relevance

Remote Laboratories have been considered as part of the *Five Major Shifts in 100 years of Engineering Education* in the Special Centennial Issue of the Proceedings of the IEEE [17], in respect to the influence of Information Communications and Computational Technologies.

In particular, the interest on federation of remote laboratories is growing. The Labshare project survey [18], made on all 34 Australian universities offering undergraduate engineering programs, reflects that the interviewed executives were more

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

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

[3] http://labshare-sahara.sf.net

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 euros 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 European wide federation and use of remote laboratories and virtual experimentations for learning and teaching purposes"*[4]. Indeed, the IP project Go-Lab[5], funded by this call with 10 million euros, aims to support a wide federation of remote and virtual laboratories. Parallel and related efforts have been placed on systems that index remote laboratories located at different institutions such as lab2go [19] or even grant access to laboratories as LiLa [20], [21], [22].

III. Existing Integration of Remote Laboratories in Learning / Content Management Systems

The relevance of this field is that, as detailed in [23], there are several services duplicated between remote laboratories and learning management systems. The administration and user experience would increase if they were merged. Both systems usually support user authentication, authorization, group management, administrative tools, user tracking, and even scheduling. Some integration approaches suggest to delegate all these services to the LMS, but some of these services will still be at least shared, such as scheduling (especially when federation systems arise) or user tracking (since some interactions with the remote laboratory might occur outside the scope of the web browser).

There are two types of solutions in the literature for using Remote laboratories in C/LMSs: ad-hoc solutions or based on federation protocols.

A. Ad-hoc Solutions

In order to integrate remote laboratories and LMSs, [3] discusses the usage of SCORM and [21] implements an architecture around it. This technology is designed to be supported by different LMSs and indeed multiple LMSs have implemented different versions. However, since it is a client-side technology and therefore it cannot contain any server code, it does not support a secure way to exchange credentials, ensure reservations or return results to the LMS.

Other approach is to develop an ad-hoc plug-in to include a particular remote laboratory on a LMS, applying the required changes in the remote laboratory [24]. This approach is common in the literature, and sometimes it is implemented by just copying or exchanging the users among both systems. Within the field of integrating remote laboratories on electronic tools appears the integration of remote laboratories on CMSs. In [25], the remote laboratory relies on Joomla to perform all the administrative tasks. This approach is interesting since it does not duplicates all the tasks refered in [23]. However, it is an example of an ad-hoc integration which does not support the integration of other remote laboratories neither the integration on other CMSs.

B. Federated Solutions

In [1], it was proposed other approach to integrate remote laboratories in C/LMSs, relying on the existing federation protocols already existing in certain RLMSs. The advantage of this approach is that indeed, federation protocols have the same targets as the integration protocols of remote laboratories in C/LMSs, where the consumer system is the C/LMS. Indeed, if in the description of federations presented in Section II-C, the term *consumer system* was replaced by *C/LMS*, the following description would be presented, which applies very well to the definition of integration of remote laboratories in C/LMSs:

- The C/LMS manages the authentication and authorization of its students.
- The provider system manages the scheduling and the access to the laboratories, storing what the users did.
- The C/LMS will later ask for results to the provider system.
- In every moment, the provider system does not need to know anything related to the particular students.

So as to demonstrate this, [1] presented two use cases, both using the WebLab-Deusto RLMS and using plug-ins which consumed the federation protocol for two C/LMSs: Joomla and Moodle. In both cases, the development was dependent on the federation protocol of WebLab-Deusto, and two C/LMS plug-ins which required some administration tasks (for managing the authentication and authorization) were required. The use of federation for integrating different tools has also been addressed in the literature [26].

IV. Discussion

As it was already presented in [1], it is possible to use federation protocols to integrate remote laboratories in C/LMSs. However, the solution presented in that contribution was only suitable for those particular systems (WebLab-Deusto, Joomla, Moodle). However, RLMSs need to support multiple C/LMSs, since they are installed in multiple universities which may use different C/LMSs.

If 3 C/LMSs were aimed, and 2 RLMSs were attempted to be supported, then 6 integrations (and testing) would be required. This does not only increases the time required to support other systems, but also decreases the chances of doing such implementations until it is well known that somebody is going to require them.

Another problem is that these C/LMSs plug-ins shared certain logic. Concepts such as *course A and course B have permission on laboratory L* needed to be implemented in all of them. The implementation and maintenance of the federation protocols of the RLMSs was also required, and it is not a trivial task.

However, let us assume that there was a new component in the middle, between RLMSs and C/LMSs. C/LMSs would build plug-ins aiming this middle component, instead of focusing on a particular RLMS. This component might have a plug-in system, and RLMSs would do the work of developing plug-ins for it, instead of focusing on particular C/LMSs.

[4] http://cordis.europa.eu/fp7/ict/telearn-digicult/telearn-objectives_en.html

[5] http://www.go-lab-project.eu/

The result would be that only 3 C/LMS plug-ins would be required for supporting all the RLMSs that had developed plug-ins. In the same way, 2 RLMS plug-ins would be required for supporting all the C/LMSs. Using this approach, if a RLMS is deployed in a location where a fourth C/LMS was used, it would be easy that they developed the support for that C/LMS. The interesting point is that once they do it, all the supported RLMS (present and future) benefit from this. In the same way, if a RLMS aimed to be supported using this system, it would suddenly be available for all the C/LMSs supported by this component.

Furthermore, once the component supports other existing protocols, RLMSs can be used by them automatically. For example, several LMSs support IMS Basic LTI, which is a standard for interoperability among Learning Tools. Once the component supported Basic LTI, RLMSs could be used by those LMSs supporting Basic LTI without any plug-in or change in the LMS, as it is evaluated in the next section. This approach also opens new opportunities. The GLUE! architecture [27] supports that certail tools (where gateway4labs could be a tool) are consumed by other systems, such as Mediawiki or LAMS. While not tested, it is theorically possible to connect both architectures in a seamless way to benefit each other and extend their tools to be connected.

The target of this contribution is precisely to propose and demonstrate that it is possible to develop such a canonical component, and this improves the current state of integrations of remote laboratories in C/LMSs.

V. Evaluation

So as to evaluate, such component has been implemented and released under an Open Source license, and it has been called gateway4labs[6]. A number of plug-ins to support C/LMSs and RLMSs have been implemented. This section describes this component and how this addresses the contribution of this statement and analyzes it.

A. Overall design

The architecture of gateway4labsis described in Figure 2. In the middle, there is a component called *LabManager*. Then, there are two parts:

- The C/LMS side or *consumer side*, where the different C/LMSs implement one of the supported protocols. The LabManager is not aware of the particular type of C/LMS which consumes it.
- The RLMS side, or *provider side*, where the different RLMSs are managed. The LabManager has a single plug-in for each RLMS, and the RLMSs remain unchanged, since their federation protocols are used.

The LabManager has a registry per each C/LMS, and it is aware of what courses are in each of them. The LabManager administrator configures what laboratories are available in each RLMS, and what C/LMSs have permissions on which laboratories. Then, when an administrator of the C/LMS system enters in the LabManager, he can select which courses will get permissions to each laboratory available for that C/LMS. Finally, students enrolled in those courses are automatically enabled to use those laboratories.

The LabManager is Open Source so it can be deployed in each institution, but also the RLMS side can install it to enable access to those universities interested in consuming their laboratories.

B. Consumer side

In the C/LMS side there are two supported approaches: IMS LTI or plug-ins in the C/LMSs by using a RESTful HTTP interface.

The *LabManager* implements IMS LTI (Learning Tools Interoperability, a standard developed by IMS), so any C/LMS implementing it can consume the *LabManager* and therefore, all the laboratories supported by those supported RLMSs. At the time of this writing, this includes Moodle since its 2.2 release, as well as Blackboard, Sakai, WebCT among others. This way, all these LMSs will automatically support all the supported RLMSs directly, without implementing any type of plug-in.

However, not all the C/LMSs support IMS LTI. Ilias, dotLRN, older versions of Moodle still used in production, and most CMSs do not support it. For this reason, gateway4labsalso provides a simple bidirectional HTTP interface that can be consumed by other systems.

The following is a sample request which the C/LMS must perform to the *LabManager*, stating that there is a valid user called "john" (which neither the *LabManager* or the RLMS know), authenticated in the C/LMS, who is requesting the laboratory identified as "robot" for that particular C/LMS. This student is enrolled in a set of courses as student.

```
POST /gateway4labs/labmanager/requests/ HTTP/1.0
Authorization: Basic dW5lZDpwYXNzd29yZA==

{
   "user-id"     : "john",
   "full-name"   : "John Doe",
   "is-admin"    : true,
   "user-agent"  : "Mozilla/5.0 (X11; Ubuntu; ...",
   "origin-ip"   : "192.168.1.1",
   "referer"     : "http://.../",
   "courses"     : {
        "01"     : "s",
        "02"     : "s",
        "03"     : "t", // "t" = teacher, "s" = student
        "04"     : "s",
   },
   "request-payload" : "{
        'action'      : 'reserve',
        'laboratory'  : 'robot',
   }"
}
```

The *LabManager* relies on the C/LMS for guaranteeing this information. With it, it can check the permissions of those courses to see whether the user can use the "robot" laboratory or not, and this laboratory has previously been mapped by the *LabManager* administrator to a particular RLMS. These courses have previously been added by the C/LMS administrator in the LabManager, using other HTTP interface, but this time provided by the C/LMS through a plug-in.

It is important to remark that this protocol is between the C/LMS server and the *LabManager*. This way, the information

[6] http://github.com/gateway4labs

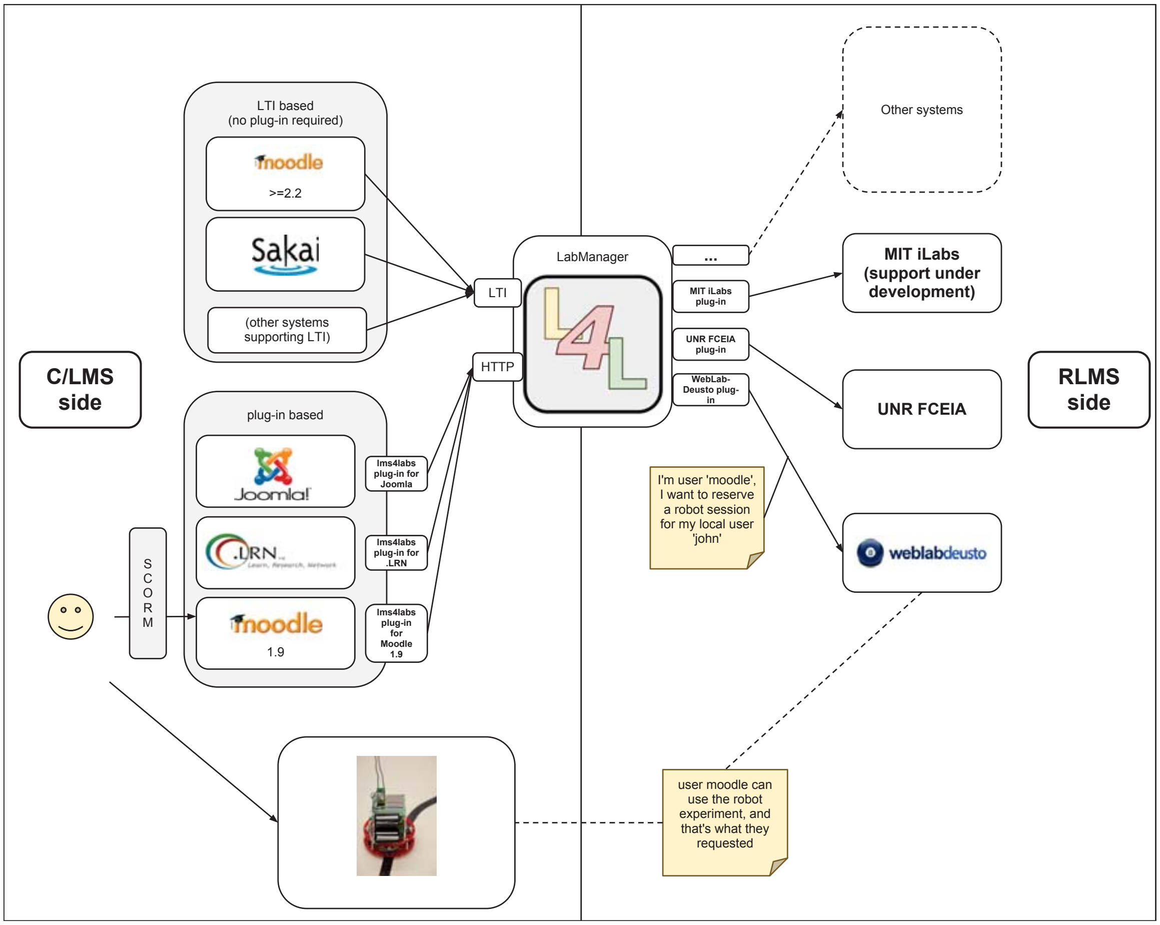


Fig. 2. Architecture of gateway4labs

is securely submitted, but it is required to modify the C/LMS (usually through a plug-in) to do this.

To make the development of this task easier in the case of LMSs, it is optionally possible to rely on SCORM. SCORM is a standard well supported in LMSs which enables instructors to upload contents in a generic format. This format does not support server side code and therefore any solution only relying on it will be insecure. However, the LMS plug-in could implement a simple HTTP API defined by gateway4labs, which enables SCORM objects to be shared among different LMSs supporting gateway4labs. This way, the SCORM objects call this HTTP API, which is implemented by the particular LMS plug-in and calls the HTTP interface explained above at the *LabManager*. This way, the LMS plug-in remains small, since even the contents are developed by the instructors. Unfortunately, for CMSs or systems not supporting SCORM, the user interface must be implemented from scratch.

The IMS LTI approach has been tested with Moodle 2.3, without requiring any development at LMS level (no plug-in was required). The HTTP interface approach has been tested with two LMSs (Moodle 1.9 and dotLRN, both using the SCORM approach mentioned above), and with one CMS (Joomla, without using SCORM).

C. Provider side

The provider side refers to the part of the *LabManager* which supports the different RLMSs. At the time of this writing, a simple interface has been defined, and two implementations have been developed on top of it. The interface basically defines operations such as: *what laboratories do you provide?* and *perform a reservation to a user called "John" to the "Robot laboratory"*, relying on the RLMS federation protocol. The RLMS trusts the *LabManager* and does not need to know "John". The RLMS generates a URL which is passed to the user through the *LabManager*, and therefore the user will finally go to the RLMS with the scheduled session.

At the time of this writing, this interface is always requesting an access *now*, so it fits very well with queueing systems, where students are granted URLs which make students wait in the queue until they use it. However, it has not yet been defined its counterpart for calendar-based booking.

The two plug-ins developed for supporting RLMSs are WebLab-Deusto and FCEIA-UNR. WebLab-Deusto [28] is a pure RLMS on top of which multiple remote laboratories have been developed, mainly in the University of Deusto (Spain) but also in other universities. FCEIA-UNR [29] is a remote laboratory developed in the Universidad Nacional de Rosario (UNR, Argentine).

Support for MIT iLabs is still under development at the

TABLE I. SUPPORTED C/LMSS AND LINES OF CODE. ADMINISTRATION PAGES LINES OF CODE INCLUDED.

C/LMS	Type	Approach	Lines of code
Moodle 2.3 (and others)	LMS	IMS LTI	0
Moodle 1.9	LMS	HTTP + SCORM	504
dotLRN	LMS	HTTP + SCORM	708
Joomla	CMS	HTTP	676

TABLE II. SUPPORTED RLMSS AND LINES OF CODE.

RLMS	Lines of code
WebLab-Deusto	890
FCEIA-UNR	141

time of this writing.

D. Results analysis

This subsection analyzes the approach presented, mainly from the point of view of the complexity and flexibility of the system.

First, let us analyze what is required to support a new C/LMS. In the case of those supporting IMS LTI, nothing is required. As seen on Table I, supporting Moodle 2.3 costs zero lines of code, since no plug-in is required. In the case of those systems where a plug-in is required, the system does not need to deal with authorization (which courses can access which laboratories is implemented in the *LabManager*), neither with the particular laboratories (the plug-ins for the RLMSs are implemented in the *LabManager*), but only implementing the HTTP interface. Additionally, it is possible to support the HTTP interface for supporting SCORM objects and therefore the same SCORM objects can be shared among different LMSs. As seen on Table I, implementing this requires really few lines of code (and take into account that code for administration pages is included).

Second, let us analyze what is required to support a new RLMS. A plug-in must be developed in Python in the *LabManager*, and it will be automatically used by all the C/LMSs. This plug-in depends on the complexity of the RLMS interface. Table II shows that it still requires few lines of code in the case of WebLab-Deusto and FCEIA-UNR.

These results show that a small amount of code is actually required for both C/LMS and RLMS support. This is so since the *LabManager* already implements some of the key shared features, such as user interfaces and selecting which RLMSs are available for each particular course (see Figure 3).

Third, it is remarkable that using this design, every component is implemented only once. The FCEIA-UNR plug-in is developed using Python in the *LabManager* and it is automatically available for all the C/LMS, present and future, regardless they are implemented on PHP (Joomla, Moodle), TCL (dotLRN), Java (Sakai, which supports IMS LTI) or any other technology. No re-implementation is required. This should encourage RLMS developers to develop and provide their own plug-ins to support a wide range of C/LMS systems.

Fourth, as explained in [30], in the RLMS side, depending on its complexity, it is possible to achieve more complex chains. For example, WebLab-Deusto supports transitive federation (University A sharing a laboratory to University B automatically enables University B to re-share it with University C). Applied to this case, this means that being the *LabManager* in University C still enables students to access laboratories of University A.

Finally, from the point of view of efficiency, it might seem that so many layers add an notable latency. However, while the reservation process might add this latency due to the different networks that could be involved (user → C/LMS → *LabManager* → RLMS), the interaction with the final RLMS is direct, so the added latency once using the laboratory is zero.

VI. CONCLUSIONS AND FUTURE WORK

This contribution is based on the assumption that it is possible to integrate remote laboratories in C/LMSs using federation protocols [1]. On top of this assumption, it shows how it is possible to use this approach collecting several common features (e.g., authorization or federation protocols clients) into a single software component. This way, instead of multiplying the supported C/LMSs by the RLMSs for calculating the number of integrations (e.g., supporting 4 C/LMSs in 3 RLMSs would require 12 ad hoc integrations), the number of integrations are notably decreased (e.g., supporting 4 C/LMSs in 3 RLMS would only require 4 integrations with the middle component plus 3 RLMS plug-ins, which not only sums only 7, but each integration is much smaller). Additionally, this encourages different RLMS developers to work together in the support of C/LMSs that will be automatically supported for them.

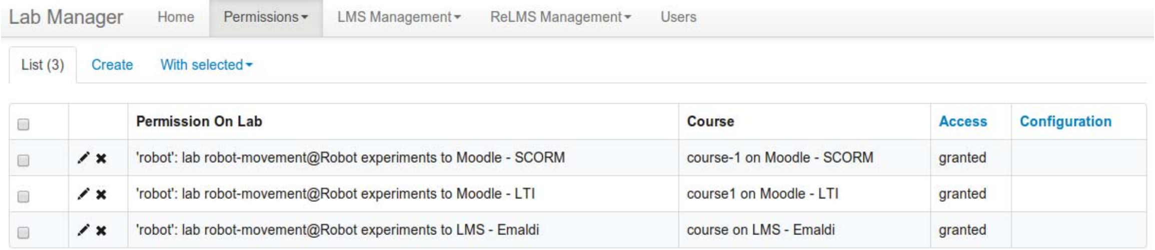
So as to demonstrate it, this middle component has been implemented, as well as support for certain RLMSs (WebLab-Deusto, FCEIA-UNR) and C/LMSs (using both IMS LTI – which includes Moodle since 2.2 and Sakai among others–; as well as through plug-ins in Joomla, Moodle 1.9 and dotLRN). The number of lines of code have been presented to verify that once the middle component (called *LabManager*) is implemented, all these integrations become small, therefore reusing a lot of code (the whole *LabManager*). This implementation is open source and available for being tested[7].

Regarding future work, ongoing efforts on polishing the system target to use it in production in different universities with students. There are other efforts to support other approaches in the consumer side, to not only support C/LMSs but also Personal Learning Environments (PLEs) such as Graasp[31], as well as other architectures that connect different tools, such as GLUE! [27]. Additionally, one of the drawbacks of the current provider API is that it only supports requests for accessing the remote laboratory at the time of the request, instead of also supporting a calendar-based booking scheme. With its support, more RLMSs will be supported. Finally, the system does not support usage retrieval at this stage.

ACKNOWLEDGMENT

This paper is supported by the Basque Government project FUTURE INTERNET II.

[7] http://github.com/gateway4labs



□		Permission On Lab	Course	Access	Configuration
□	✎ ✖	'robot': lab robot-movement@Robot experiments to Moodle - SCORM	course-1 on Moodle - SCORM	granted	
□	✎ ✖	'robot': lab robot-movement@Robot experiments to Moodle - LTI	course1 on Moodle - LTI	granted	
□	✎ ✖	'robot': lab robot-movement@Robot experiments to LMS - Emaldi	course on LMS - Emaldi	granted	

Fig. 3. Example screen of the Lab Manager

REFERENCES

[1] P. Orduña, E. Sancristobal, M. Emaldi, M. Castro, D. L. de Ipiña, and J. Garcia-Zubia, "Modelling remote laboratories integrations in e-learning tools through remote laboratories federation protocols," *2012 Frontiers in Education Conference Proceedings*, vol. 0, pp. 1–6, 2012.

[2] O. Dziabenko, J. García-Zubia, and I. Angulo, "Time to play with a microcontroller managed mobile bot," in *Global Engineering Education Conference (EDUCON), 2012 IEEE*. IEEE, 2012, pp. 1–5.

[3] L. Gomes and S. Bogosyan, "Current trends in remote laboratories," *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 12, pp. 4744–4756, 2009.

[4] C. Gravier, J. Fayolle, B. Bayard, M. Ates, and J. Lardon, "State of the art about remote laboratories paradigms-foundations of ongoing mutations," *iJOE*, vol. 4, no. 1, 2008.

[5] B. Carisa, A. Burain, S. Molly H, and C. Lawrence, "Running control engineering experiments over the internet," 1995.

[6] B. Aktan, C. Bohus, L. Crowl, and M. Shor, "Distance learning applied to control engineering laboratories," *Education, IEEE Transactions on*, vol. 39, no. 3, pp. 320–326, 1996.

[7] J. Henry, "Running laboratory experiments via the world wide web," in *ASEE Annual Conference*, 1996.

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

[9] R. Cedazo, F. Sanchez, J. Sebastian, A. Martínez, A. Pinazo, B. Barros, and T. Read, "Ciclope chemical: a remote laboratory to control a spectrograph," *Advances in Control Education–ACE*, vol. 6, 2006.

[10] J. Del Alamo, L. Brooks, C. McLean, J. Hardison, G. Mishuris, V. Chang, and L. Hui, "The mit microelectronics weblab: A web-enabled remote laboratory for microelectronic device characterization," in *World Congress on Networked Learning in a Global Environment, Berlin (Germany)*, 2002.

[11] D. Gillet, H. Latchman, C. Salzmann, and O. Crisalle, "Hands-on laboratory experiments in flexible and distance learning," *Journal of Engineering Education*, vol. 90, no. 2, pp. 187–191, 2001.

[12] I. Gustavsson, J. Zackrisson, L. Håkansson, I. Claesson, and T. Lagö, "The visir project–an open source software initiative for distributed online laboratories," in *Proceedings of the REV 2007 Conference, Porto, Portugal*, 2007.

[13] Z. Nedic, J. Machotka, and A. Nafalski, "Remote laboratory netlab for effective interaction with real equipment over the internet," in *Human System Interactions, 2008 Conference on*. IEEE, 2008, pp. 846–851.

[14] R. Safaric, M. Truntič, D. Hercog, and G. Pačnik, "Control and robotics remote laboratory for engineering education," *International Journal of Online Engineering (iJOE)*, vol. 1, no. 1, 2005.

[15] F. Torres, F. Candelas, S. Puente, J. Pomares, P. Gil, and F. Ortiz, "Experiences with virtual environment and remote laboratory for teaching and learning robotics at the university of alicante," *International Journal of Engineering Education*, vol. 22, no. 4, pp. 766–776, 2006.

[16] J. Hardison, K. DeLong, P. Bailey, and V. Harward, "Deploying interactive remote labs using the ilab shared architecture," in *Frontiers in Education Conference, 2008. FIE 2008. 38th Annual*. IEEE, 2008, pp. S2A–1.

[17] J. Froyd, P. Wankat, and K. Smith, "Five major shifts in 100 years of engineering education," *Proceedings of the IEEE*, vol. 100, no. 13, pp. 1344–1360, 2012.

[18] T. Kotulski and S. Murray, "The national engineering laboratory survey," *Labshare Project. December*, 2010.

[19] D. Zutin, M. Auer, C. Maier, and M. Niederstatter, "Lab2go – a repository to locate educational online laboratories," in *Education Engineering (EDUCON), 2010 IEEE*. IEEE, 2010, pp. 1741–1746.

[20] T. Richter, Y. Tetour, and D. Boehringer, "Library of labs-a european project on the dissemination of remote experiments and virtual laboratories," in *Multimedia (ISM), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 543–548.

[21] 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.

[22] V. Mateos, A. Gallardo, T. Richter, L. Bellido, P. Debicki, and D. Villagrá, "Lila booking system: Architecture and conceptual model of a rig booking system for on-line laboratories," *International Journal of Online Engineering (iJOE)*, vol. 7, no. 4, pp. pp–26, 2011.

[23] E. Sancristobal, M. Castro, J. Harward *et al.*, "Integration view of web labs and learning management systems," in *Education Engineering (EDUCON), 2010 IEEE*. IEEE, 2010, pp. 1409–1417.

[24] J. Ferreira, A. Cardoso *et al.*, "A moodle extension to book online labs," *International Journal of Online Engineering*, vol. 1, no. 2, 2005.

[25] M. Abdulwahed and Z. Nagy, "Developing the trilab, a triple access mode (hands-on, virtual, remote) laboratory, of a process control rig using labview and joomla," *Computer Applications in Engineering Education*, 2010.

[26] P. Orduña, P. H. Bailey, K. DeLong, D. López-de Ipiña, and J. García-Zubia, "Towards federated interoperable bridges for sharing educational remote laboratories," *Computers in Human Behavior*, 2013.

[27] C. Alario-Hoyos, M. L. Bote-Lorenzo, E. Gómez-Sánchez, J. I. Asensio-Pérez, G. Vega-Gorgojo, and A. Ruiz-Calleja, "Glue!: an architecture for the integration of external tools in virtual learning environments," *Computers & Education*, 2012.

[28] P. Orduña, J. Irurzun, L. Rodriguez-Gil, J. Garcia-Zubia, F. Gazzola, and D. López-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.

[29] F. Lerro, S. Marchisio, and O. Von Pamel, "Exploring didactic possibilities of an electronic devices remote lab with students of electronic engineering," in *CD Proceedings of the International Conference on Interactive Computer aided Blended Learning ICBL2009*, 2009.

[30] P. Orduña, F. Lerro, P. Bailey, S. Marchisio, K. DeLong, E. Perreta, O. Dziabenko, I. Angulo, D. López-de Ipiña, and J. Garcia-Zubia, "Exploring complex remote laboratory ecosystems through interoperable federation chains," in *Education Engineering (EDUCON), 2013 IEEE*. IEEE, 2013.

[31] E. Bogdanov, C. Ullrich, E. Isaksson, M. Palmer, and D. Gillet, "From lms to ple: a step forward through opensocial apps in moodle," in *Advances in Web-Based Learning-ICWL 2012*. Springer, 2012, pp. 69–78.