



Interoperating remote laboratory management systems (RLMSs) for more efficient sharing of laboratory resources



David Lowe^a, Herbert Yeung^a, Mohamed Tawfik^{b,*}, Elio Sancristobal^b, Manuel Castro^b, Pablo Orduña^c, Thomas Richter^d

^a Faculty of Engineering and IT, University of Technology, Sydney (UTS), Australia

^b Electrical and Computer Engineering Department (DIEEC), Spanish University for Distance Education (UNED), Spain

^c The Internet Unit of the Deusto Institute of Technology—DeustoTech, University of Deusto, Spain

^d The Multimedia Department for Research and Education of the Computing Center of the University of Stuttgart (ITSUS), Germany

ARTICLE INFO

Article history:

Received 14 July 2013

Received in revised form 16 June 2015

Accepted 31 July 2015

Available online 8 August 2015

Keywords:

Distance learning

Interfaces

Services architectures

Standards and interoperability

ABSTRACT

In this paper we propose a novel Application Programming Interface (API) design pattern for inter-communication between Remote Laboratory Management Systems (RLMSs) accommodating different levels of functional support and thereby allowing more efficient sharing of laboratory resources regardless of their hosting RLMS. Afterwards, we present initial results and demonstrate the feasibility and effectiveness of this pattern by developing an API for two common RLMSs, Sahara and the iLab shared Architecture (ISA). As a result, users logging into a Sahara server managed to access and manipulate a radio-activity experiment hosted on an ISA server.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Laboratories are generally recognized as important tools for education and scientific research. The challenges in resourcing conventional laboratories [1] and the rapid evolution of computer technology and communication networks have supported the emergence of remote laboratories as a viable educational tool [2,3]. As a result, remote laboratories have strongly been adopted in many engineering disciplines, especially those oriented to control and industrial real world, such as industrial informatics [4–7] and industrial electronics [8,9]. Remote laboratories have become increasingly sophisticated, with a growing body of research considering aspects such as flexibility of access [10], ability to share resources and labs [11,12], security of users, data, and devices [13], and accessibility for disadvantaged users [14] among many others [15]. To a significant extent, many of these issues have been successfully overcome, with continuous, reliable and high quality services being maintained for much of the past decade.

Cross-institutional laboratory sharing is one of the most often raised justifications for the use of remote labs [11,16]. This can lead to improved utilization levels, shared costs, and access to a much broader

range of laboratory apparatus. Thus, the focus of remote laboratory development is now moving towards more sustainable models that promote both institutional and individual engagement [16]. Rather than individual academics custom building equipment for their specialized subjects, remote laboratory development is increasingly being carried out by multi-institution consortia [16]. Despite its claimed benefits, the related initiatives have been historically very limited and struggled to achieve sustainability. Early examples of attempts to support shared access RLMSs include the World Wide Student Laboratory [17] and PEARL [18] projects. Similarly, the LearNet [19], ProLearn [14] and PEMCWebLab [20] projects were carried out during the early to mid-2000s with little success. More recently the LiLa [21], Sahara [11,16,22,23], ISA [24–26], Weblab-Deusto [27,28], NANSLO (Western Interstate Commission for Higher Education 2012) and UniSchools [29] projects have continued to attempt to support laboratory sharing.

A key element of these developments is the creation of Remote Laboratory Management Systems (RLMSs) that provide a common online framework for accessing and administrating a wide pool of heterogeneous developed remote lab systems that might be distributed across different institutions and geographical locations [13]. They provide services such as booking, assessment, tracking, and synchronous and asynchronous communication tools, as shown in Fig. 1.

RLMSs should be agnostic with regard to the remote laboratory design in order to support the widest range possible of remote laboratories. Examples of RLMSs that are specifically designed to directly manage a collection of remote laboratories include iLab Shared Architecture (ISA)

* Corresponding author.

E-mail addresses: david.lowe@uts.edu.au (D. Lowe), Herbert.Yeung-1@uts.edu.au (H. Yeung), mtawfik@ieec.uned.es (M. Tawfik), elio@ieec.uned.es (E. Sancristobal), mcastro@ieec.uned.es (M. Castro), pablo.orduna@deusto.es (P. Orduña), richter@rus.uni-stuttgart.de (T. Richter).

[24–26], and Sahara [11,16,22,23]. Other systems have some related functionality but have a different focus. For example, LiLa [12] provides some laboratory services (such as location and booking services) but not others (such as monitoring the laboratory status), though it does extend into wider support for the experimental activities. Each of the different RLMSSs has typically arisen from a different set of pedagogic and technical design parameters, and philosophical approaches. Each has different strengths and weaknesses, and potentially addresses a different set of needs. We argue that this diversity is an asset that should be encouraged rather than avoided. However the diversity does represent a challenge with regard to supporting cross-institutional sharing of laboratories between institutions using different RLMSSs.

Currently, for a remote laboratory that is provided by one institution to be accessed by users from a different institution, typically those users must directly access either the laboratory itself or the providers RLMSS (if one exists). This can represent a significant hurdle to effective sharing given that it potentially places a substantial burden on the provider of the laboratory to coordinate access for the users. It can also mean that students who access multiple laboratories (possibly shared from multiple providers) will need to manage numerous access accounts and system interfaces.

Common issues facing RLMSS could be summarized as follows:

- Inability to achieve Inter-institutional sharing and interoperability with heterogeneous RLMSSs in order to share learning objects (ex., courses, remote labs, etc.) among institution.
- Lack of a standard design pattern, which increases the developing time for attaching the same remote laboratory to different RLMSSs at different institutions.
- Inability to access labs from different systems within a single interface as well as duplicated maintenance efforts for accounts on multiple systems.

This can be partially addressed by an architecture that uses federation and single sign-on technologies so that a student from institution A can access the RLMSS from institution B through a mechanism defined by the federation and contracts between institution A and B. The downside of this is that the students must still access different remote laboratories through different systems. A common portal might address this to some extent, but the system would then potentially lack a consistent look-and-feel. An alternative, which we focus on in this paper, is to allow the home institution of the students to maintain their own RLMSS, and for this system to be able to interoperate with other RLMSSs

so that students can directly access provider's experiments. In other words, consider the scenario shown in Fig. 2, where the RLMSS ISA is installed at the Massachusetts Institute of Technology (MIT) and providing access for the MIT users (MIT 1, MIT 2...) to experiments installed at MIT (MIT Exp A, MIT Exp B...). On the other hand, the RLMSS Sahara is installed at the University of Technology, Sydney (UTS), providing access for the UTS users (UTS 1, UTS 2...) to experiments installed at UTS (UTS Exp A, UTS Exp B...). Adopting an Application Programming Interface (API) to communicate between both systems, it would be possible for a UTS user to gain access through Sahara to experiments installed at MIT and integrated in ISA if there is an agreement between both institutions and as long as both architectures support this API in order to interconnect with each other. Since API is meant to be language independent, any RLMSS provider could implement them even if the RLMSS architecture was not initially considering its provision in its designed. For instance, it could be provided as a plug-in. It is also necessary to mention that the API approach provides a solution for a higher level communication among RLMSSs, and once the communication is established and access to a particular remote lab is granted, communication will be dictated by the lab itself.

In this paper we address this limitation and describe a novel API design pattern for inter-communication between RLMSSs defining the underlying design principles, the supported profiles and the set of calls underlying each profile, as shown in Section 2. In the same section, we explain the capabilities of the proposed design using a sequence of illustrative scenarios. Afterwards, in Section 3, and basing on the proposed design pattern, we develop and implement an API to allow a radio-activity experiment hosted on the ISA server at the University of Queensland to be accessed by users logging into a Sahara server at the UTS. Finally, the conclusion and the future works are addressed in Section 4.

2. Design pattern

The proposed pattern was developed by the faculty of Engineering and IT at UTS with the aid of several partners from the Global Online Laboratory Consortium (GOLC) [30], which encompasses pioneers in remote laboratory development from all over the world. In this section the pattern is defined and discussed. However, it is anticipated that as the builders of RLMSSs gain experience with this pattern further refinements will be required. The pattern was realized from a wide perspective in order to allow developing APIs that are adaptable with any existing

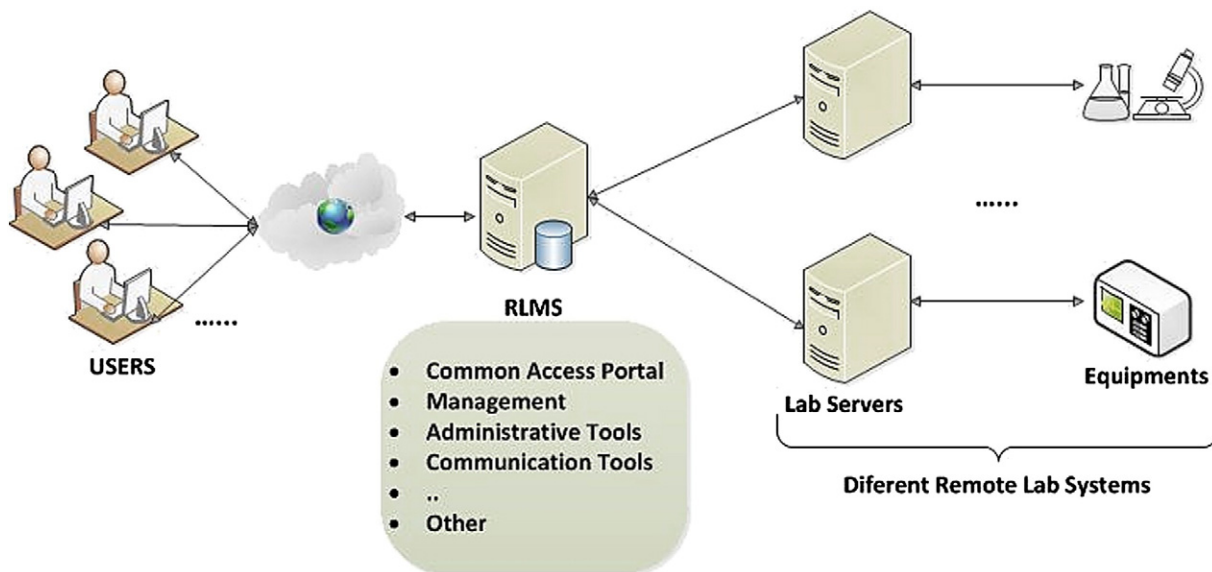


Fig. 1. Architecture of RLMSS.

Download English Version:

<https://daneshyari.com/en/article/453480>

Download Persian Version:

<https://daneshyari.com/article/453480>

[Daneshyari.com](https://daneshyari.com)