



Challenges and solutions in remote laboratories. Application to a remote laboratory of an electro-pneumatic classification cell



Miguel A. Prada*, Juan J. Fuertes, Serafín Alonso, Sergio García, Manuel Domínguez

Suppress Research Group, Universidad de León, León, Spain

ARTICLE INFO

Article history:

Received 23 December 2014

Received in revised form

2 March 2015

Accepted 3 March 2015

Available online 11 March 2015

Keywords:

Remote laboratories

e-Learning

Control education

Rich internet applications

Open standards

ABSTRACT

The field of remote laboratories has been developed over the years. These laboratories are valuable educational tools that allow students and faculty interact with real equipment through the Internet, as if they were physically in front of the system. When a remote laboratory is developed, many technical difficulties are found, mainly with respect to the links between the different elements such as physical system, database and clients. In this work, we analyze the challenges and propose approaches to develop remote laboratories that achieve flexibility, scalability and greater educational value. In particular, we present a remote laboratory for automatic control, which includes an electro-pneumatic classification cell. The architecture of the laboratory is based on the three-tier architecture (physical system layer, server layer and client layer). The client layer has been implemented with web standards such as HTML5, AJAX and CSS3, the clients interact with the system using a content management system and the communication interface between the physical system and the database uses a middleware based on OPC (Ole for Process Control). Engineering students have regularly used the remote laboratory and they have evaluated it through surveys.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

A complete learning experience in engineering education should include experimentation with real systems to introduce professional practice and skills, support analytical concepts and increase the involvement of the students (Lindsay & Good, 2005). Remote laboratories have proven to be effective educational tools in engineering, with comparable results to traditional on-site labs (Nickerson, Corter, Esche, & Chassapis, 2007). Indeed, they provide additional advantages, such as their flexibility, which is appealing for the students. The remote laboratories also enable sharing the scarcely available equipment. This is a relevant issue because education on technological subjects often requires the use of complex and expensive equipment. As a result, the development of remote laboratories in the field of automatic control has advanced significantly in the last years (Guzmán et al., 2010; Leva & Donida, 2008; Vargas et al., 2010).

The field of remote laboratories has achieved a certain level of maturity. Nowadays, several platforms provide much more than the simple remote control/monitoring of a physical system through the Internet. They provide resource-efficient solutions oriented to achieve aims such as scalability, platform independence, security and accessibility. There are nevertheless two main goals that any effective remote laboratory must address. One is the ability of connecting heterogeneous physical systems easily. The other one is to alleviate the workload of the website administrators, who are sometimes faculty and need to spend their time developing educational content instead of performing tasks such as user management, new systems configuration, etc. In order to provide the required level of manageability, the approaches use multi-tier architectures, centralize some core services and use standardized interfaces for communication.

Next, some relevant examples of remote laboratories are presented. It can be seen that some of them manage the communication with the system completely, storing data and controlling the interaction, whereas other platforms simply set up a connection and do not participate in the data exchange. We are more interested in the platforms that support completely managed experiments.

* Corresponding author.

E-mail addresses: mapram@unileon.es (M.A. Prada), jj.fuertes@unileon.es (J.J. Fuertes), saloc@unileon.es (S. Alonso), sergio.gdeprado@unileon.es (S. García), manuel.dominguez@unileon.es (M. Domínguez).

A platform supporting managed and unmanaged experiments is the WebLab-Deusto remote laboratory. This remote laboratory is a framework developed to decouple infrastructure development and experiment development (García-Zubia, Orduña, López-de Ipiña, & Alves, 2009). Through plug-ins and APIs available in different client-side and server-side technologies, they achieve certain aims such as scalability or distribution. Regarding infrastructure, the architecture uses a login server for authentication and a core server to manage scheduling, storage, requests, etc. The laboratory and experiment servers are used to manage the laboratory. The unmanaged experiments can use, for instance, virtual machines. In the experiments that are managed by the platform, the communication is made through a set of commands using web services. The clients need to use the client library, with methods for submitting a command, whereas the experiment servers can implement the server library or use XML-RPC directly.

Another example of comprehensive remote laboratory is the iLab Shared Architecture (ISA) of the Massachusetts Institute of Technology (MIT), which follows a typical three-tier structure. This distributed software toolkit pursues scalability, platform independence and efficient management (Hardison, DeLong, Bailey, & Harward, 2008). The first tier is the user application, available in a web browser. The second tier is called the service broker, which provides authentication and manages the access to the experiments. The third tier comprises the lab servers that are connected to the physical devices and controls the experiment. The lab client and the lab server only communicate directly to the service broker. For that purpose, two APIs exposed as web services with pass-through methods are available. Each experiment needs a lab server and the clients must be able to produce and interpret XML documents in compliance with a lab client/server communication schema. The lab servers also need to interpret and produce those documents and transform experiment parameters to physical commands. The architecture can be shared between two or more institutions. In this case, the two first tiers are located in an institution and the broker server communicates with the lab server in charge of the experiment.

On the other hand, the University of Technology, Sydney (UTS) Remote Laboratory follows an unmanaged approach. The architecture of the UTS remote laboratory is developed to be flexible, extensible and able to manage multiple sets of equipment (Lowe, Murray, Lindsay, & Liu, 2009). The user needs a web browser and a remote desktop client. When the user starts a session on a web browser, the arbitrator allocates equipment to the user, provides visual monitoring of the experiment and boots a virtual machine on a master server to enable control of the system. To control the system, the user creates a remote desktop connection to this virtual machine. The arbitrator supports access queues and management of access to multiple identical systems.

As can be seen from the overview of available platforms, there are 3 key issues that need to be addressed in the design of a remote laboratory. First, it is necessary to design the communication interface with the control systems of the physical resources. This interface must enable easy configuration of existing and new equipment in the platform. Second, we need to establish the functionality and structure of the core server, sometimes also known as arbitrator or broker. This server, or set of servers, is in charge of authentication, user and session management, resource allocation, etc. and therefore it is crucial in the operation of the laboratory. Third, a client-side communication policy must be designed, because this choice influences the development of the clients.

In the last years, we have worked on this 3 key issues through the development of the Remote Laboratory of Automatic Control (LRA-ULE). It is a laboratory oriented to research and education with a focus on real industrial equipment. The LRA-ULE allows the engineering students of subjects related with control systems working with both simulation and teaching or industrial equipment. Moreover, this platform is flexible and scalable, so it is possible to add new control systems from different manufacturers easily.

In this paper, we review the challenges and propose solutions to implement remote laboratories. In particular, we present approaches to facilitate the management of the platform, the communication between the different layers of this platform and the client-side functionality. These approaches are implemented in the LRA-ULE remote laboratory and tested by means of the incorporation of an educational flexible manufacturing system to the LRA-ULE. The system consists of an automated classification system developed with steel profiles, electro-pneumatic actuators and a robot with six degrees of freedom (DOF). The system includes several elements, such as a programmable logic controller, two variable frequency drives and different sensors and actuators, all of them commonly used in real industrial systems. This way, the students can study a flexible manufacturing system, analyze its components and monitor the operation of each element. Previously, some authors have proposed remote laboratories for flexible manufacturing cells (Bellmunt, Miracle, Arellano, Sumper, & Andreu, 2006; Castilla, Guzmán, Moreno, & Rodríguez, 2009; Reynard, Gomis-Bellmunt, Sudria-Andreu, Boix-Aragones, & Benítez-Pina, 2008), focusing on programming of PLCs (programmable logic controllers) and the SCADA (supervisory control and data acquisition) system.

This paper is structured as follows: In Section 2, the main challenges posed by remote laboratories and their possible solutions are presented. The communication interfaces proposed for the remote laboratory platform are described in Section 3. After that, Section 4 explains the usage of the network architecture to access through the web to the flexible manufacturing system. Section 5 describes in detail some proposed practical tasks and explains the results of the questionnaires answered by the students to validate and improve the laboratory. Finally, conclusions are discussed in Section 6.

2. Challenges and solutions in remote laboratories

One of the objectives pursued by any remote laboratory is the seamless integration of physical equipment and control systems, regardless of the characteristics or manufacturer. This challenge can be addressed through the joint application of the *three-tier architecture* and *open standards*. The three-tier architecture is a client-server solution, commonly used in business web development, which, as shown in Section 1, has been implemented with small differences in several remote laboratories (Harward et al., 2008). The implementation in the remote laboratory described in this paper includes a physical layer, a middle layer and a client layer (Domínguez, Fuertes, Prada, Alonso, & Morán, 2011). The physical layer interfaces with the physical systems that are shared via Web, their data acquisition system and their controllers. The middle layer provides the shared common functions, i.e., data storage, user access policies, and management of the control system. Finally, the client or presentation layer is the application offered to the end users. Communication between layers is accomplished by means of well-known open standards.

Another challenge is the ability to focus on the development of educational content. The great value of a remote laboratory comes not from the availability of many systems or simulations on-line, but from the educational contents linked to those resources and the learning value they offer to the students. However, management and maintenance of the remote laboratory structure is often a burden. These tasks

Download English Version:

<https://daneshyari.com/en/article/348265>

Download Persian Version:

<https://daneshyari.com/article/348265>

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