Using ISA services to manage lab sessions with embedded lab servers

Danilo G. Zutin
Carinthia University of Applied Sciences
Villach, Austria
dgzutin@ieee.org

Tiago Faustino Andrade
University of Porto
Porto, Portugal
tfa@fe.up.pt

Maria Teresa Restivo
University of Porto
Porto, Portugal
trestivo@fe.up.pt

Manuel Rodrigues Quintas
University of Porto
Porto, Portugal

***Abstract*—The implementation and maintenance of online laboratories is expensive and partially associated with high administrative efforts. These are reasons, why a common framework capable of efficiently managing online laboratory resources, users and providing mechanisms to create and maintain online laboratory sessions is necessary. An important aspect in the implementation and deployment of an online laboratory is the schema used to assign laboratory time to users. It can be either a queuing based system or a calendar-based booking. While this may sound trivial, the decision between one and another can change how the system is implemented. In this paper we will outline how we will deploy the services provided by the iLab Shared Architecture to manage the access to a pool of embedded lab servers hosted by the University of Porto.**

Keywords—online services; remote monitoring; remote sensing; booking system; remote experiment

I. Introduction

Remote laboratories can be considered resources that enable remote access to a specific piece of equipment. Such resources have to be assigned to users during an experiment execution or laboratory session.

Basically remote laboratories can be classified into two different types: Batch and interactive.

Batched experiments run asynchronously. According to [1] this means that the entire course of an experiment execution is specified before the experiment starts. The lab session consists of submitting an experiment protocol, executing the experiment, and then retrieving and analyzing the results. Typically, batched experiments run quickly so that scheduling is rarely necessary and a queuing mechanism is in most cases sufficient to manage the access to the remote piece of equipment. One of the reasons for that is that the experiment protocol specification and results analysis can be done offline by the user client.

In the other hand, interactive experiments are those that run synchronously. This means that a user session has to be maintained during the entire time slot that comprises the experiment execution. Interactive experiments usually require more time to run and might require as well that the hardware is exclusively available to user running the experiment.

Interactive experiments require therefore a completely different model in comparison to the batched ones to assign resources to its users.

According to [2], queuing mechanism are mostly used in batched laboratories, but interactive labs can implement different methods of scheduling, most typically queuing, or a calendar-based booking. For this work we will use the iLab Shared Architecture (ISA) to manage access to lab servers running on embedded devices hosted by FEUP.

A. The OnlineLab @ FEUP

One of the most universal solutions for remote labs is based on LabVIEW software, Laboratory Virtual Instrument Engineering Workbench, later in nineties from the time when LabVIEW offered the web server functionality [3, 4] as well as also happened with MatLab/Simulink. Since then developed works were reported from many different places and applications, e.g. [5-7]

LabVIEW has been also the most popular solution used at the OnlineLabFEUP. Each instrumented experiment and its LabVIEW application communicate through hardware interfaces from National Instruments - I/O data acquisition cards of USB or PCI type. Moodle platform is used to integrate each self-contained learning object as well as the access to the “Remote Experiment” and all the essential documentation: “Experiment Description”, “How to use the experiment” and “Evaluation topic”. It also integrates the booking application lying between Moodle platform and the LabVIEW user interface [8].

A Microsoft IIS main webserver contains all the information on the available experiments and a database for authentication purposes – however, this functionality is not working any more due to the lack of upgrading in the version of the booking interface and this is at the present one big constrain. The PHP application of the booking interface integrated in the Moodle platform is still used to schedule the experiment. In the main server the video web server is also running.

In the lab, the approach is “one experiment – one web server” being this supplied by LabVIEW from National Instruments. When the booking allows the user to access the experiment s(he) will access directly its server and so the system set-up through hardware interfaces.

The booking system provides time slots of 1 hour. A pre-defined interval tolerance is used in order to make the slot free if the user is not in time. Also, any of the experiments does not need more than 15 min. So, it was added to each experiment (when the control is granted to the user by the experiment web server) a small application named “time out” which allows the user to use the experiment for 15 min long. When this "time out" is over, the system application is automatically closed and a new user is allowed to go in, [8].

This architecture is not sustainable for developing new systems relying on it. To some of the referred constrains the need of maintenance is also a problem – often the systems do need to be rebooted.

The need of a completely new solution has been clear and it became possible as a consequence from other developments.

So, due to the interest in the use of microcontrollers from Microchip the present solution was implemented using those microcontrollers and it has been tested in terms of robustness.

II. Booking Solution

As previously mentioned, ISA will be used to provide the necessary scheduling services to manage access to the online laboratories at FEUP.

ISA is a software architecture developed at the MIT (Massachusetts Institute of Technology) that facilitates a cross institution sharing and management of online labs. ISA provides a framework for the maintenance of a lab session, lab users management and experiment data storage. It establishes clear rules governing the communication between clients and their respective online lab servers by means of an API (Application programming Interface) based on Web services SOAP calls. ISA distinguishes the tasks of using a specific lab that comprises an experiment from the tasks of managing users’ accounts, user authentication and other tasks that follow a lab session. This clear separation of roles is one of the main advantages of this software architecture. ISA does not focus in a specific type of laboratory but provides a set of general purpose functions for lab developers [8]. ISA is divided into three tiers that provide different services. These tiers are client, Service Broker and lab server. The Service Broker is the core of the architecture and provides user authentication, authorization, experiment data storage and access to scheduling services. The lab clients and lab servers are lab domain specific.

ISA is a distributed software architecture that facilitates sharing Online Labs across institutions. Its design had considered some requirements like authentication, authorization, group management and experiment data storage. Beyond these requirements, the architecture should support the use of a diverse number of laboratory hardware and software and should not tie client and server platforms. It should also not make any assumptions on the network policies (firewalls, proxy servers) that a user might be under.

Among the services provided by an ISA installation is the *scheduling service*. It consists of two separated Web services, the User-side Scheduling Service (USS) and the Lab-side Scheduling (LSS).

Only the Service Broker can authenticate a user and determine if he/she has the right to schedule a lab session. The user uses the USS to schedule a session for a lab, based on the available time slots published by the LSS. The LSS is responsible for setting the policies for a specific lab and can run with different USS’s and lab servers. The policies mentioned might be a lab setup, etc. In order to use the ISA scheduling services, an Online Lab must be built in a way that scheduling for a specific user will be ensured [9]. Fig. 1 depicts the interactive services of ISA and how these services interact to establish a lab session.

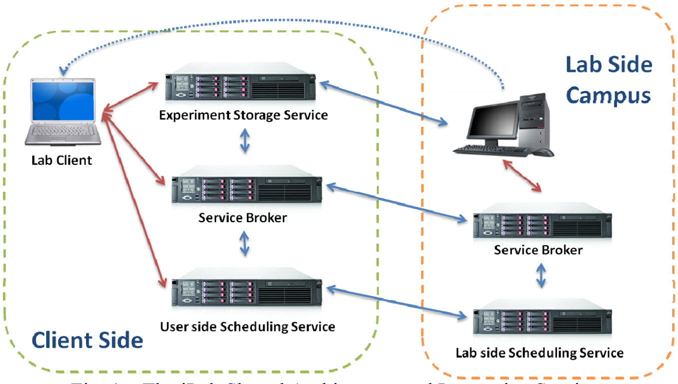

Fig. 1 – The iLab Shared Architecture and Interactive Services

The process involving the booking of a lab session in ISA is orchestrated by the service broker, lab-side scheduling and user-side scheduling services as previously described. Assuming that a reservation of a time slot for a specific online laboratory is done an authorization mechanism is responsible for verifying the permissions of a user to access a specific lab. The scenario involving an interactive experiment the level of complexity is much higher as other services are involved in the authorization process. The authorization was implemented in the form of ticketing. With this mechanism the service broker issues tickets that represent the permission of a user or agent to access a resource. When attempting to access an online laboratory the user presents a coupon ID that is used by the lab server to retrieve an associated ticket against the service broker. Once the lab server confirms the validity of the ticket, the access to the laboratory is granted to the user. These credential information is added by the service broker to the request when the user is redirected to the lab server, therefore this process is completely transparent to the user.

A typical scenario would be the following. As soon as a student logs in and is authorized to execute an experiment, tickets are created along with a coupon representing the collection. The coupon is passed to the lab client. In order to connect to the lab server, the client sends the coupon to the lab server that retrieves than the ticket with the Service Broker. If a valid ticket is returned, the user is authorized for the amount of time reserved to use the lab. Also for storing the data with the ESS, a ticket coupon is passed to the ESS, which retrieves the data storage ticket. Ticketing is also based on Web services and must be provided by the developer of the lab server [9].

III. HARDWARE SOLUTION

The microchip provides several development systems. In this work the PICDEM.net 2 Development Board was used with free source code from the TCP/IP Stack version 5.42. This complementary tools accelerates the development of systems where is need the integration of Ethernet connectivity.

The PICDEM.net 2 includes several features. However only some of them are highlighted in this work:

- The microcontroller installed on the board is the PIC18F97J60 which manage all surrounded peripherals and with a built-in Ethernet controller and transceiver;
- A serial EEPROM communicates by the Serial Peripheral Interface BUS (SPI) with the microcontroller and provides 256 Kbits for storage the web pages. It is also used to storage nonvolatile configuration options such as the IP address, the mask, the default gateway, etc.;
- The LCD with 16 columns by 2 lines is very useful for displaying error messages and for helping the debugging code;
- The interface components such as LEDs driven by digital I/O pins of the microcontroller were used to simulate a digital output;
- The push buttons also connected to digital I/O were used to simulate a digital input;
- The potentiometer is directly connected to one of the analog input.

The architecture of the TCP/IP Stack is modular in design (divided into multiple layers) and it is written in the 'C' programming language. Each layer accesses services from one or more layers directly below it. Microchip TCP/IP Stack support several protocols such as: ARP, IP, ICMP, UDP, TCP, DHCP, SNMP, HTTP, FTP and TFTP. It also includes other features: Secure Sockets Layer (SSL), NetBIOS Name Service and Domain Name System (DNS) [10].

IV. DEVELOPED EXAMPLES – SHORT DESCRIPTION

Two systems with different goals were developed based on the PICDEM.net 2. The first one is a monitoring temperature system and it was developed to test the new solution as a concept proof system. The second one is related to an experiment as a demonstrator and a tutorial of the topic "encoder working principle", being the tutorial the remote system.

A. *Monitoring temperature system*

To test the robustness of an embedded web server from Microchip in supporting a remote experiment, a very simple system for temperature monitoring was implemented. The system monitors two temperatures. One is the room temperature in the lab being the other the focus's temperature of a halogen lamp. The option for this type of lamp has to do with getting a fast and pronounced changes in temperature, this is to get a “dynamic” experiment even involving temperatures.

An integrated circuit from MAXIM was used for the temperature's measurement which delivers an output data signed in 14 bit for the thermocouple (used to follow changes in the halogen lamp temperature) and 12 bit for the internal solid state sensor (following room temperature).

Fig. 2 shows the webpage that embeds the user interface of the “monitoring temperature system”.

B. *Remote experiment based on the encoder principle*

Robotics, CNC machines, nuclear processes, wind turbines, telescopes, process lines, labelling, drilling and mixing machines and medical equipment are some of many application areas for encoders where they are used as feedback sensors in motor speed control, as sensors for measuring and positioning, etc.

Optical encoders are electrical mechanical transducers that generate digital signals as a consequence of motion its moving part of linear or rotary type. A sensor head (integrating LED and phototransistors) reads “a code printed in bar or disk coupled to the moving body” arranged in different ways leading to optical encoders of absolute or incremental type [11].

Both rotary encoders, responding to rotation, and linear encoders, which respond to motion in a direction, are available. The rotary solutions are the most universal. In fact, when used in conjunction with mechanical conversion devices, rotary encoders can also be used to measure linear movement, speed, and position.

Incremental encoders generate series of pulses and absolute encoders generate a digital word. In the first case pulses can be used to measure speed or the position by using a counter. Absolute encoders generating digital words indicate absolute positioning.

Considering the relevance of their use and the short comprehension students always have on this topic a remote tutorial has been designed to be used either as a demo kit or as a remote experiment.

The solution for this experiment is centered on the use of two microcontrollers. The master microcontroller is the installed on the PICDEM.net 2 and allows users to interact and communicate with the remote experiment and manage the interaction between the user and the experiment. The second microcontroller supports the encoder and the components needed to the experiment.

The use of the two microcontrollers is looking to have a modular systems working either as a local portable experiment or also its remote experiment.

Fig. 3 shows a functional demonstrator prototype of the "encoder working principle" that will be integrated into the new remote laboratory.

Fig. 2 – Monitoring temperature system webpage

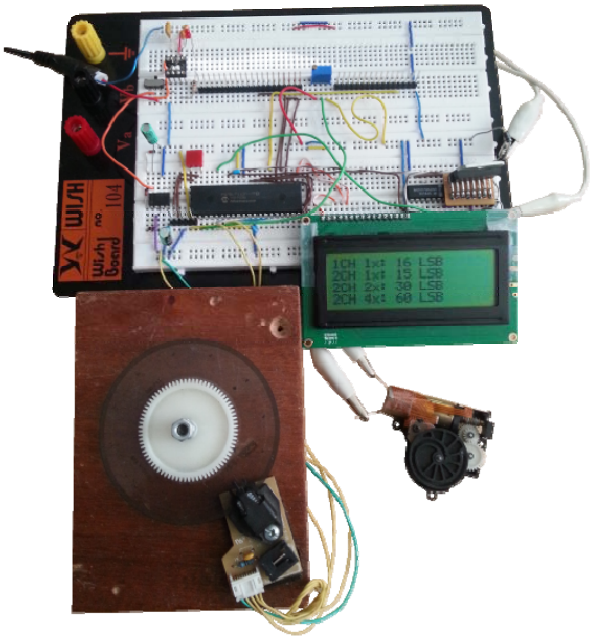

Fig. 3 – Encoder kit being assembled

C. *Next planed demos*

Load cells are force transducers and they are based in mechanical bodies instrumented with electrical strain gauges. The mechanical body of a load cell, when submitted to a force, deforms strain gauges bonded to it which convert the deformation to electrical signals.

Load cells are used in many applications as in automotive, aerospace, oil and gas industries, in lever scales, mechanical weighing, scales manufacturing, in medical applications, etc. There are load cells of many different types and ranges.

Considering the relevance of being universally used, a particular load cell will be designed and a kit will be developed in order to demonstrate how to use as a scale in a theoretical lecturing room and later it will be remotely available for students training.

LVDT is an electromechanical transducer which converts a displacement into an electrical signal of novel characteristics. This transducer is also, itself, an interesting example of a amplitude modulator system. Because of novel characteristics it is used in many fields: civil engineering, mechanical structures, automation machinery, power generation, machine tools, R & D activities, etc.

Due to the relevance of its use and also due to the fact that students do not identify easily many of these features, one LVDT will be integrated in a kit to perform demonstrations at the theoretical lecturing room and later it will be remotely available for students training.

V. Conclusion

This paper proposes the use of the ISA services to manage the access to the embedded lab servers of FEUP. Beyond the advantages of providing a booking system for the described embedded remote lab solutions presented the implementation of ISA brings other advantages, namely the possibility of sharing and of making these online labs deployable via a scalable distributed architecture. Scalability can be defined as the capacity of a system to support a growing load of users and in terms of the network, the ability of it to grow without extra efforts. It offers user management functionalities and if the laboratories deployed grow in terms of number of lab installations and lab users no changes have to be done.

Online labs have been growing in number lately, however the authors do not see it as a replacement to traditional hands on labs but rather as an additional resource that opens up different and new possibilities, namely, collaborative work among dispersed peers, sharing of labs in a cross-institutional basis, in class demonstrations. Different institutes and schools could share experiments and knowledge in a collaborative manner that parallels real-life working conditions. Importantly, online labs can be also used in workplace settings where there is a pressing need to apply these systems to continually provide learning opportunities for workers who must adapt to rapidly changing conditions.

The system presented in this paper is however based on embedded Webservers that do not support the use of a server-side scripting language that could interact with the web services API and implement the iLab lab server web services API it is necessary to use an intermediate server capable of implementing these tasks. However, it is important to mention that this is a work in progress that can change during the course of its implementation.

Acknowledgment

The authors would like to express special thanks to the UNITE Program and Dr. Bogdan Deaky.

References

[1] V. J. Harward, J. A. del Alamo, S. R. Lerman P. H. Bailey, J. Carpenter, 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, June 2008.

[2] D. Lowe, N. Orou. "Interdependence of Booking and Queuing in Remote Laboratory Scheduling ", REV 2012 Conference, Bilbao, Spain.

[3] Integrating the Internet into your Measurement System, National Instruments, March 1999 Edition.

[4] D.G. Alexander and R.E. Smelser,. "Delivering an Engineering Laboratory Course Using the Internet, the Post Office, and a Campus Visit", Journal of Engineering Education, Vol. 92, No. 1, 2003, pp. 79–84, 2003

[5] Auer, M.E.; Pester, A.: Interaktive Lehrmaterialien mit MATLAB, Proceedings of the ICL2000, Villach/Austria, 2000

[6] A.V. Fidalgo, R.J. Costa, J.M. Ferreira, G.R. Alves, "Experimenting the 1149.1 and 1149.4 test infrastructures in a Web-accessible remote Lab (without Plug-ins!)", XVI Conference on Design of Circuits and Integrated Systems (DCIS'01b), Porto, Portugal, 2001.

[7] M.T. Restivo, J.G. Mendes, C.M. Silva and C. Sousa, "A Remote Laboratory @ Feup – First Steps", World Congress on Engineering and Technology Education, pp. 97-100, Brazil, 2004.

[8] J.M. Ferreira and A.M. Cardoso, "A Moodle extension to book online labs," Proceedings of Remote Engineering and Virtual Instrumentation International Symposium, REV'05, 2005.

[9] Hardison, J. L., DeLong, K., Bailey, P. H and Harward, V. J. 2008. Deploying Interactive Remote Labs Using the iLab Shared Architecture, 22-25 October 2008, NY, Saratoga Springs.

[10] Microchip TCP/IP Stack Help Copyright (c) 2012 Microchip Technology, Inc. All rights reserved.

[11] M.T.Restivo, F.G.Almeida, M.F.Chouzal, J.G.Mendes and A.M. Lopes, "Handbook of Laboratory Measurements and Instrumentation", International Frequency Sensor Association (IFSA) Publishing, Barcelona, May 2011, ISBN: 978-84-615-1138-9. http://www.sensorsportal.com/HTML/BOOKSTORE/Handbook_of_Measurements.htm.