Networking Online Labs Within the ISA Framework

doi:10.3991/ijoe.v5i4.1084

Danilo Garbi Zutin
Carinthia University of Applied Sciences, Villach, Austria

***Abstract*—The proposed paper focuses in outlining the implementation of online laboratories within the framework of the iLab Shared Architecture (ISA) in order to facilitate sharing those labs among several institutions and managing laboratory users. The laboratories are compliant with the ISA batched architecture, which is based on Web services to mediate the communication between Lab Servers and Lab Clients. The first laboratory presented is a Hybrid Lab that allows for the realization of real experiments as well as simulations with an analog programmable device.**

***Index Terms*—online laboratories, Web services, remote control of devices.**

I. INTRODUCTION

Using online laboratories has the potential of significantly reducing obstacles related to cost, time-inefficient use of facilities, inadequate technical support and limited access to laboratories. Nowadays there exist a large number of online lab solutions, but expenditure is growing and the life cycle of online lab software and hardware is becoming shorter.

It is still difficult to share instrumentation and experiments among laboratories. Each one has its own security policy and adopts a proper technology in accessing and controlling real devices. A common integrated framework, offering indexing facilities, unique logins, file sharing and the seamless access and run of experiments, is the main challenge in order to create a network of online laboratories. Grid technologies can be used to set up an effective network of remote laboratories for education purposes by sharing instrumentation and resources. However, the evolution of remote laboratories from the current client/server architecture to grid-based architecture requires well-defined tools for location, security, and integration of resources, and further research is currently being conducted to examine this issue [13-15].

As Universities and other institutions are likely to develop their own solutions and standards to deliver online laboratories to their users, no trend to a unique standard is observed, creating an obstacle against sharing these online labs. Considering the current scenario, a migration towards standardized solutions for delivering online labs becomes necessary to ensure software reusability and therefore facilitate online labs development and sharing.

At this point, the ISA (iLabs Shared Architecture) comes into play. 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 user's 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.

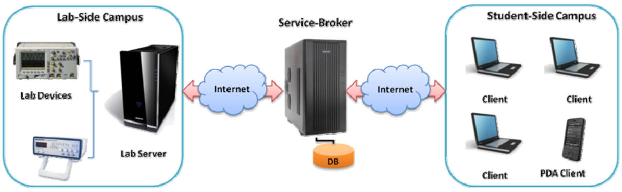


Figure 1. The iLab Shared Architecture

ISA proposes a classification for online experiments. On one side there are the batched experiments and on the other the interactive experiments. Batched experiments are those in which all parameters necessary to run it are specified before execution. On interactive experiments, the course of the execution can be changed at any time by the user. Due to their differences on the implementation, operation and technical requirements, a different architecture (ISA) exists for each one of them, although the iLab team at the MIT has already released a merged architecture in August 2008 (Fig. 1).

This work is focused on the ISA Batched experiments and proposes the implementation of two different online laboratories within this framework. In this architecture the communication between clients and laboratory is mediated by a middleware server (Service Broker), a Web application that manages users' accounts, data storage and can provide different clients with access to several different lab servers in a "many to several" mode and delegates to experiment server only the experiment execution.

The interface that provides the communication between clients with service broker and service broker with lab servers is implemented with Web services, and is therefore platform independent. That means that clients and lab servers can be developed in any platform supporting Web services.

II. THE READ LABORATORY

The Laboratories proposed to be developed within this work are a lab with an analog programmable device and a lab developed on top of the NI ELVIS platform and based on the iLab ELVIS lab from the MIT. This laboratory allows users to test several different circuits.

The first experiment previously mentioned is called READ – Remote ASIC Design and Test and was developed with an Analog ASIC (Application-Specific Integrated Circuit) and allows users to remotely test and program the device over the Internet as well to create different circuits with an application software and upload them to a real device.

It's a so-called hybrid lab that combines design and simulation as well as real hardware control and test.

The device under test used is an ispPAC10 from Lattice Semiconductor and is capable of realizing a variety of popular analog functions including precision low-pass and band-pass filters, amplifiers, oscillators summing, differencing and integration. Its basic building block contains four integrated programmable analog modules known as PAC blocks and a programmable analog routing pool. Each PAC block emulates a collection of operational amplifiers, resistors and capacitors. The ability to program the internal capacitor values of the ispPAC10 gives the user the flexibility to vary, for example, the gain of a filter, corner frequency, and other characteristics, without the need for external components, hence allowing the designer to realize thousands of distinct analog circuits and filter characteristics from a given circuit architecture.

The READ system has already been up and running since 2 years. The efforts now employed to adapt this lab for the iLabs shared architecture are justified by the possibilities of sharing it with other institutions in an easier and most efficient way.

This online lab can be implemented as a batched experiment and therefore we decided to include it to the ISA batched architecture.

The circuit under test was assembled into a mockup board that provides a connection to the data acquisition card terminals and a JTAG interface that allows the circuit to be re-programmed on the fly by the user.

The diagram of Figure 2 describes the READ Lab Server, which was based on the MIT ELVIS lab. The Web services API as well as the Process Manager are modules internal to the Web server. The first one exposes the Web methods accessed by the Service Brokers that are registered for this Lab Server. The Process Manager validates experiments and queues in a database that works as a gateway for the communication between the modules internal to the Web server and the Experiment Engine.

The Experiment Execution Engine is an independent and highly domain-specific module of the Lab Server as its code is strongly dependent on the nature of the experiment. The experiment engine connects the Lab Server software with the lab hardware. It communicates with the modules internal to the Web Server (ASP.NET application) through a database that implements a queue for the jobs that should be executed.

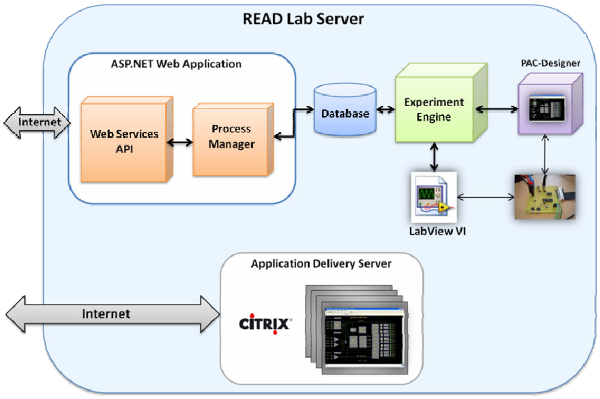


Figure 2. The READ Lab Server

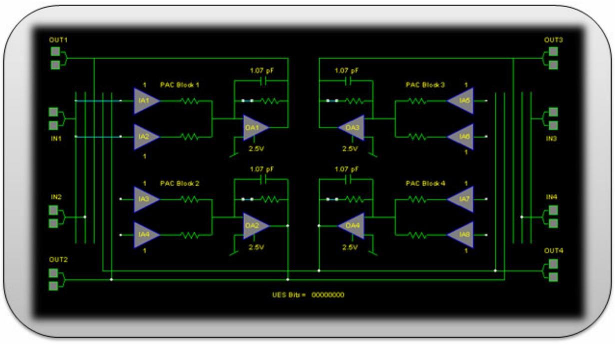


Figure 3. PAC-Designer Software Interface

A LabVIEW virtual instrument compiled as a dll controls the data acquisition to perform the measurements and another module that uses the PAC-Designer libraries to upload the desired schematics into the device. As this is a batched experiment we must ensure that the proper schematic is uploaded before the measurements are taken. The batched experiment specification and results use XML as a vehicle to exchange information between lab servers and lab clients and therefore both sides must be able to interpret these XML files.

A. The PAC-Designer Uploader

The ispPAC device can be programmed with the PAC-Designer tool. This software is free and can be downloaded from Lattice Semiconductor's website. Although, any assumptions should be made about the user's Internet connection and network polices that might not allow the installation of any software. In order to free the user from downloading any application we decided to make the PAC-Designer software available via a Presentation delivery server (Citrix Server) delivered by means of a Java Applet client.

The PAC-Designer user interface is shown on Figure 03. With this software the user can design the schematics for the device, perform simulations and save the designed circuit locally in his/her machine. The saved file is send within the experiment specification as an extra parameter to be submitted by the lab client to the batched architecture. When the Lab Server receives and parses the experiment specification, the .PAC file will be recovered and handed out to the software module that uploads it to the device. This software module was written in Visual Basic.NET by using the PAC-Designer Software Development Kit.

B. The READ Lab Client

The functionalities implemented by the Lab Client are the following:

1. Provide the lab Graphical User Interfaces
2. Include pedagogical aspects
3. Implement the Web Services interface to communicate with the Service Broker
4. Create experiment specification protocols
5. Parse experiment results received from the server

The client communicates with the READ lab server and is also delivered as a Java Applet, as this is a type of client supported by the iLab shared architecture. This client communicates with the service broker by means of the Web services interface and use the pass through methods of the service broker API to articulate the communication with the lab server. With the client the user can specify the batched parameters listed below:

1. Waveform Type
2. Frequency
3. Amplitude
4. Offset
5. FG Node connection
6. OSC Node connections
7. Time Base
8. Channel
9. Coupling
10. Type of measurement

These parameters are wrapped by the experiment specification XML and sent to the lab server via the service broker, which does not understand the meaning of the forwarded parameters or the nature of the experiment being carried out.

C. *The READ Lab Client User Interface*

The user interface provides the tools for visualization and interaction of the user with the experiment. Figure 4 shows the user interface of the lab client. The lab client developer should consider some pedagogical aspects and use familiar notation (commonly used in real laboratories).

The GUI (Graphical User Interface) was programmed with the Swing editor of Netbeans IDE. The interface organizes separately the controls in four main fields, the function generator, the oscilloscope, the PAC-Designer Uploader and the graph. The batched parameters are transferred to the program through text fields and combo boxes. These parameters are sent to the main module that builds the experiment specification XML. The field for the PAC- Designer uploader contains a button "Browse" and a text field that shows the PAC file to be uploaded.

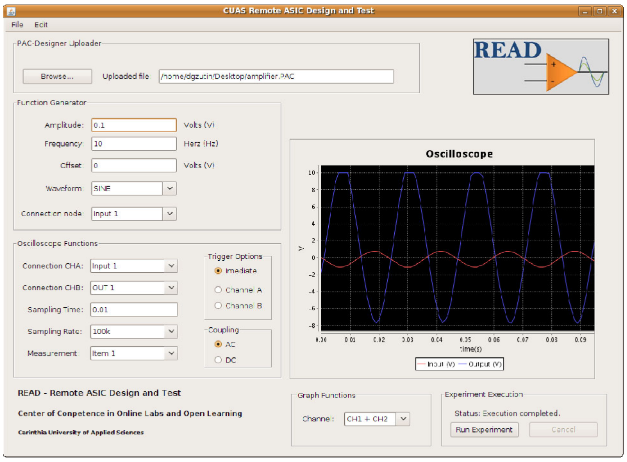
III. THE ELVIS LABORATORY

The second online laboratory to be described by this paper uses a set of fixed electronic circuits and was built upon an iLab ELVIS server and client developed by the MIT.

We have extended its functionality by developing some hardware modules and a main board that hosts these modules. Figure 05 shows the main board and the modules. The first version of this hardware is connected directly to the NI ELVIS prototyping board and on Figure 04 two circuit modules can be seen attached to the main board, although it will be able to accommodate up to 6 of these modules.

This lab provides a set of electronic circuits implementing several well known and typical electronics experiments with operational amplifiers. Some of these circuits are an inverting and non inverting amplifier, an integrator, an adder, etc. The circuits were assembled on a printed circuit board specially designed for this laboratory. The modules were designed to offer some flexibility and to increase the range of experiments supported by this laboratory.

Each one of these modules is addressed by digital I/O lines of the DAQ card and a bunch of relays is used to switch different lines connected to the oscilloscope and function generator terminals of the NI ELVIS platform. So we are able to completely isolate the circuit under test from the other parts, providing in this way a higher accuracy for the measurements performed.


CUAS Remote ASIC Design and Test

File Ecit

PAC-Designer Uploader

Browse... Uploaded file: /nome/dgzutin/Desktop/amplifier.PAC

Function Generator

Amplitude: 0.1 Volts (V)

Frequency: 10 Herz (Hz)

Offset: 0 Volts (V)

Waveform: SINE

Connection node: Input 1

Oscilloscope Functions

Connection CHA: Input 1

Connection CHB: OUT 1

Sampling Time: 0.01

Sampling Rate: 100k

Measurement: Item 1

Trigger Options: ● Imediate ○ Channel A ○ Channel B

Coupling: ● AC ○ DC

READ - Remote ASIC Design and Test

Center of Competence in Online Labs and Open Learning

Carinthia University of Applied Sciences

Graph Functions

Channel: CH1 + CH2

Experiment Execution

Status: Execution completed.

Run Experiment Cancel

Figure 4. READ Lab Client User Interface

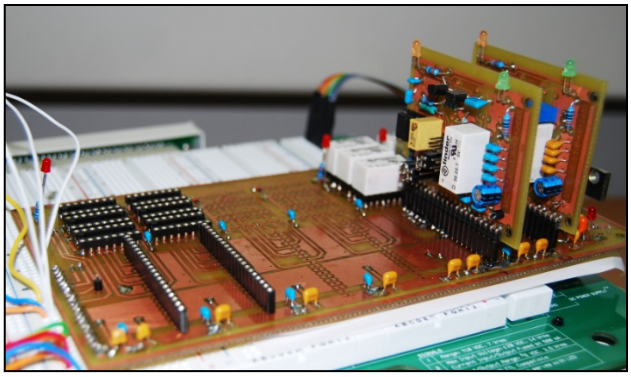


Figure 5. Circuit modules developed for the iLab ELVIS Lab

The user selects the desired circuit on the client application and submits the parameters to the server.

As previously mentioned, the iLab batched architecture supports Java Applet and HTML clients. Both lab clients previously mentioned are implemented as Java Applets. These applets embody the user interfaces and provide graphs to display the measurements taken from the circuits under test. The user can save the results and analyze them later using his/her preferred software.

For the experiment with the analog ASIC a completely new client was developed, while for the ELVIS lab a client developed at the MIT was used and modified to include new parameters.

Each module seen on Figure 04 also contains an extra relay that gives the possibility to modify the circuit under test (adding a capacitor in parallel with the feedback resistor, for example) and potentiometers are used to control the gain of the circuit without the necessity to replace components.

The DAQ (Data Acquisition) card used for both online laboratories was the NI PCI 6251. It provides sampling rates of up to 2.86 MS/s for the analog outputs and 1.25 MS/s for analog inputs. To control the data acquisition card a LabVIEW VI (Virtual Instrument) was developed. Respecting the characteristics of batched experiments, these VIs must execute straightforward, using as inputs the parameters sent by the client and as output the measurements performed. In order to expose the LabVIEW VIs' functionalities to the Lab Server software these virtual instruments were compiled as DLLs and called from the Lab server software.

IV. CONCLUSION

Online labs can be very useful when applied to solutions involving high costs of people and equipment transportation. Online Laboratories experimenters have a great advantage because they are not subjected to limitations of locomotion, but require also a good network and a good bandwidth infrastructure to perform experiments.

By integrating online labs with the iLab Shared Architecture, different institutes and schools could share experiments and therefore knowledge in an additional manner. Also by using remote solutions it would be possible for universities with less financial resources to take advantages of expensive equipments from other institutions by means of a cooperative network of remote systems. The main goal of this work is to implement new online laboratories within the ISA framework, due to its flexibility and scalability in providing online lab developers and users a possibility to share these labs without the drawbacks usually associated to this practice.

In both presented labs there is a clear separation of the circuits under test and the technical environment. Therefore these are standard solutions, which are easy to adapt to other labs.

Our focus in this paper was to describe these two labs developed for the iLabs batched architecture, due to the characteristics of these labs, but not all experiments can be specified in a batched way. Some exercises require a higher bandwidth and a direct connection between clients and lab servers. These types of experiments usually are single user systems and might require a time reservation of laboratory resources. For those labs, the use of the iLabs Interactive or merged architecture might be necessary.

REFERENCES

[1] Christou IT, Efremidis S, Tiropanis T, Kalis A (2007) Grid-Based Virtual Laboratory Experiments for a Graduate Course on Sensor Networks. IEEE Transactions on Education, USA, pp 17-26.

[2] Dick, B. Action learning and action research. http://www.scu.edu.au/schools/gcm/ar/arp/actlearn.html

[3] Faltin N, Böhne A, Tuttas, J, Wagner, B (2002) Distributed Team Learning in an Internet-Assisted Laboratory. Proceeding of the ICEE 2002, Manchester, UK.

[4] Feisel LD, Rosa AJ (2005) The Role of the Laboratory in Undergraduate Engineering Education. Journal of Engineering Education, pp 121-130.

[5] Ghercioiu M, Folea S, Monoses J (2007) The WiTAG - a WiFi Sensor TAG. ICOMP-07, Las Vegas, Nevada, USA June 25-28.

[6] Gustavsson I, Zackrisson J, Håkansson L, Claesson I, Lagö T (2007) The VISIR project – an Open Source Software Initiative for Distributed Online Laboratories. REV Conference, Porto.

[7] IT iLab URL: http://icampus.mit.edu/ilabs/ - Last visited in 11/2008.

[8] J. Harward, J. A del Alamo, iLab: A Scalable Architecture for Sharing Online Experiments, ICEE 2004

[9] Müller D, Erbe H. (2007) Collaboraive Remote Laboratories in Engineering Education: Challenges and Visions. In. Advances on remote laboratories and e-learning experiences. University of Deusto, Bilbao.

[10] National Instruments Corporation. URL: http://www.ni.com – Last visited in 11/2008.

[11] Revans R (1982) Origins and Growth of Action Learning. Bratt-Institute fur Neues Lernen, Sweden.

[12] Schmid C (2007) Grid Technologies for Virtual Laboratories in Engineering Education. REV Conference 2007, Porto.

[13] Sam Lee and Mayur R. Mehta, "Establishing a Remote Lab for Teaching Enterprise Application Development", Information Systems Education Journal, Vol. 4, No. 50, pp 1-7, August 8, 2006.

[14] James E. Corter, Jeffrey V. Nickerson, Sven K. Esche, Constantin Chassapis, "Remote Versus Hands-On Labs: A Comparative Study", 34th ASEE/IEEE Frontiers in Education Conference, Session F1G, 20–23 October 2004, Savannah, GA, USA.

[15] Qin Shuren, Bo Lin and Liu Xiaofeng, "Development of the Networked Virtual Instrument Lab for Vibration Measuring Based on Microsoft.Net", Instrumentation and Measurement Technology Conference, IMTC 2004, Como, Italy, 18-20 May 2004.

AUTHOR

D. G. Zutin is a senior researcher at the Carinthia University of Applied Sciences, Villach, Kaernten A-9500 Austria (e-mail: dgzutin@IEEE.org).

Submitted, January, 17, 2009. Published as resubmitted by the authors on May, 16, 2009.