Instruments and Experiments Control Methodology Based on IVI and LXI Technologies

http://dx.doi.org/10.3991/ijoe.v7iS2.1768

U. Hernández-Jayo, J. García-Zubía
University of Deusto, Bilbao, Spain

***Abstract*—In this paper we present a new model to control the instruments and experiments in a remote laboratory. This model is based on LAN networks and a control methodology through reusable drivers. The objective is to obtain a software control architecture independent of the hardware of the laboratory, so each institution can use its own equipments and experiments based on its needs and with minimal restrictions regarding to the hardware of the lab.**

***Index Terms*—remote laboratories, remote instrument and measurement control.**

I. INTRODUCTION

WebLab-Deusto is the concept of remote laboratory designed and developed by the homonymous research group of the University of Deusto. Since 2001 we have implemented different versions of this laboratory [1][2][3], all of them are based on the paradigm client-server. Finally, a stable version is currently available on the web site of the group under the GNU GPL license

In addition, since 2007 the Weblab-Deusto research group is partner of the VISIR project [4], in both development and deployment tasks. As a result of this project, since 2007 we are using the VISIR platform in different degrees: Telecommunication, Informatics, Industrial Technologies and Electronics. The subjects related with VISIR at this moment are: Digital Electronics, Computer Technology, Analog Electronics, Circuits and Physics.

From the experience gained during the deployment and use of the VISIR platform, we started to work in the idea of applying to the VISIR platform, one of the main concepts used in the WebLab-Deusto, this is, the independence of the software regarding to the hardware of the lab.

As we can read in the references to the VISIR project [5], [6], [7], the deployment of this platform depends directly on the use of the hardware for which it was designed: a National Instruments PXI and a switching matrix, which has been designed by the research team of the BTH, explicitly for the VISIR laboratory.

So, the method that we propose in this paper combines the best of the concept of remote laboratory proposed by the WebLab-Deusto [8], with the power of the environment designed in the VISIR project. The goal is to eliminate the need for a PXI and a proprietary switching matrix for the deployment of this remote laboratory by abstracting the software of the laboratory of the hardware.

This paper is structured as follows: Section II describes the problem that we want to solve and the proposed in this research work. Section III is focused on describing and comparing the available technologies to reach the defined goals. In Section IV the proposed solution is described, and finally at Section V the conclusions and future work are presented.

II. BACKGROUND AND OBJETIVES

One of the greatest advantages of the use of the VISIR lab as a support for teaching and from the point of view of the user is the power of its user interface. It has been developed using Adobe Flash, which allows the student to execute the same actions that would take place in the real laboratory, but in a remote way: place components in a breadboard, perform the connections between them, configure the instruments, and carry out measures over the circuit under test.

Although the use of Adobe Flash suppose a small disadvantage from a practical point of view and a drawback from the technology perspective [8] since the user needs to install the Flash Player plug-in, the great dissemination and expansion of this plug-in makes it a safe and easily deployable tool.

But from the point of view of the deployment of the VISIR lab, there are several important issues to be considered: a) the need for a PXI platform; b) to have a number of modules of the switching matrix in order to accommodate the components of the available circuits; c) to understand how the switching matrix and its management server works [9], in order to set up the matrix and describe the available circuits on the platform; d) the price both the PXI and the switching matrix.

Thus, in 2009 the WebLab-Deusto research group started to work in a new control method in order to confront and overcome these drawbacks, with the following objectives:

a) Independence the control of the instruments, in order to make possible that each institution that wants to join to the VISIR consortium would be able to use their own equipments, satisfying only a few requirements.
b) Make possible that all the instruments of the laboratory are made by a trader, without having to include any expensive and complex proprietary solution.
c) Simplify and reduce the imposed restrictions regarding to the addition of new components and experiments in the VISIR switching matrix

In addition, we want to integrate the proposed control method in the WebLab-Deusto platform in order to include in this platform laboratories not only focused on digital electronics. Besides, using this platform we can benefit from its web environment advantages: the most important one is that this management system allows the professor to obtain information on the tasks carried out by each student in each of its connections. This information is not currently available in the analogous web platform of the VISIR lab.

III. HARDWARE TECHNOLOGIES TO INSTRUMENTS AND EXPERIMENTS CONTROL

Before developing the final solution, we have conducted an analysis of the different technologies commonly used for controlling instruments and experiments in remote labs focused on electronics [10].

As important as the choice of the technologies used for the design and development of the client and the server of a remote laboratory, it is the selection of the methodology that makes possible that the actions carried out by the student in the web browser, can be executed on the physical equipment in the laboratory and once the response of the circuit is obtained it will be sent back to the client interface. For this reason, the server is in charge of controlling the instruments and the experiments needs to have a physical communication interface with the equipments and a management system to control and monitor the experiment. To this end, they are two technological alternatives (Figure 1):

a) **Modular instruments**: in this type of configuration, instead of having a user interface, a firmware and a link of communication for each instrument, the components of the control and management system are placed on a single processor and are shared by all the instruments. In this way, the instruments are embedded in a chassis or integral instrumentation platform and all of them are connected to the communications bus that puts them in contact with the processor. The most extended chassis are PXI and VXI, which used PCI and VXIBus as a bus of communications respectively. These instrumentation platforms also have communication interfaces such as GPIB or LAN, which make that they can also be controlled and monitored remotely.

b) **Standalone instruments**: in this configuration, the instruments are connected to the server as if they were peripheral. These instruments have a firmware that interprets the actions that the user performs locally on the front panels or in a remote way through the control software and the communications bus that links the remote application with the instrument. The results of these actions are displayed in the front panel of the equipment or in the user interface created in the remote control software. To control these instruments remotely, the most commonly buses are GPIB (General Purpose Interface Bus), LXI (LAN eXtensions for Instrumentation**)** and lesser extent RS-232 and USB.

Without going into extensive descriptions because they are very well known standards, the most common technologies used for the control of both types of instruments are:

a) **PXI (PCI eXtensions for Instrumentation)** The PXI systems consist of three components [11]: the chassis (in charge of providing the modularity to the system and to contain the instruments, the controller and the communication system between them), the controller (in charge of managing the whole system) and the instrumentation modules inserted into the system.

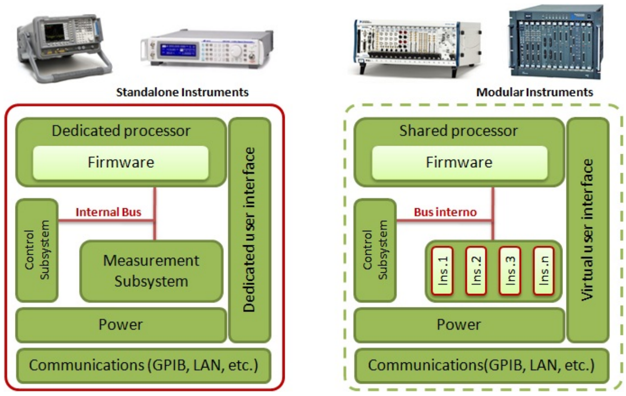

Figure 1. Instrument configuration alternatives

b) **VXI (VME eXtensions for Instrumentation)**: a VXI system generally is composed of a frame (also called mainframe chassis), which contains a maximum of 13 slots in each of which a card/instrument can be placed. The first slot (module 0), is aimed at the system control module and the remaining 12 are dedicated to modular instruments [12].

c) **GPIB (General Purpose Interface Bus or IEEE 488.2)**: GPIB is an asynchronous communication protocol, with certain limitations with respect to the hardware and the technology used for its implementation so the maximum transmission speed is 2Mbits/sec to a maximum distance of 20 meters. The devices are connected to the bus using a specific wire. The maximum number of connected instruments is 15, with a physical distance between them of 20 meters. The rule update ASE / IEEE Std.488.2-1987 defines more precisely the protocols for the exchange of messages, the data format, and general commands for the instruments. Finally, the consortium of instruments manufacturers created in 1990 to improve this update of the standard, defined the Standard Commands for Programmable Instruments (SCPI), which specifies the set of commands to program and control instruments [13].

d) **LXI (LAN eXtensions for Instrumentation)**: LXI is a standard promoted and developed by the LXI Consortium, which was created in 2004 and that in September 2005 brought to light the LXI specification. This standard is based mainly on three technologies: Ethernet, web interfaces and the Precision Time Protocol (PTP, IEEE 1588). Thus, the objective of the Consortium is to increase interoperability and functionality of the instruments that have an Ethernet interface, through the standardization of interfaces and control systems that are common to all the instruments of the same class, whether oscilloscopes, function generators, multimeters or the other instruments contained in the specification [14].

The main feature of these systems is that as they are based on Ethernet, the speed of information transmission between the devices and controllers can be very high. In addition, the progress made over the Internet in recent years has made possible for all the test and measurement equipment to incorporate this interface. In this way, likely this technology will become the most spread in the area of test and measurement equipments [15].

A. Hardware control technology selection

GPIB has been since the beginning of the 70s, the most tested and stable standard that has been used in test and measurement systems. In the area of remote experimentation, many of the laboratories that use instrumentation to carry out tests and measures on experiments, have used GPIB as control interface successfully [16] [17]. Modular instruments based on PXI and VXI also have been and they are currently used in remote labs, mainly because they are compact systems that could contain all the required instruments in the laboratory [6] [18] on a single equipment.

TABLE I includes a comparative analysis between these different technologies that can be used as control standards for instruments and experiments in a remote laboratory. This subjective analysis has been done in base of the following indicators, where each one is associated with a mark in the range 1–5:

- **Price,** this indicator reflects the costs to be considered when the control system is deployed: special or dedicated cables, cards, frames, specific drivers, etc. In this case, the highest value (5) indicates a low price, which is an advantage when it comes to any technology deployment. The lower value (1) indicates a high price.
- **Transmission speed,** this indicator refers to the maximum rate allowed by the control technology. Perhaps this indicator is not critical in certain remote labs, but it should be considered in those in which applications in real time are made, because the bandwidth could determinate the quality of service of the session.
- **Compatibility,** this indicator refers to the ability of technology to interoperate with other existing technologies.
- **Flexibility**, concerning the number of instruments that can be controlled by the system and the distance between the control system and the equipments and experiments.
- **Longevity** refers to the capacity of the technology to be included as control interface in future instruments or it is going to be an obsolete technology in a near future.
- **Ease of use** of the technology by the user and the developer
- **Size**, refers to the size of the equipments and the limitations regarding to special requirements: temperature control, humidity, power, etc.

Analyzing numerically the results shown at TABLE I, the following can be deduced that LXI is numerically the most valued technology and looking at the most important aspects, LXI is also more valued. PXI and VXI are mainframe systems that could offers specific solutions to specific problems, but the newest systems as LXI can offer similar solutions but providing a bigger bandwidth and speed connections. Moreover, these systems are based in parallel communication bus, so the location of the equipments is limited due to the distance between them. One solution is to use a GPIB bus, but it has the limitation of the number of equipments connected to the instruments network, and also the distance between them. However, as LXI is based on Ethernet network, the distance between the equipments and the bandwidth are not a problem.

TABLE I
CONTROL TECHNOLOGIES FOR REMOTE LABS ANALYSIS

	PXI	**VXI**	**GPIB**	**LXI**
Price	1	1	3	4
Transmission speed	4	4	2	5
Compatibility	2	2	5	5
Flexibility	2	2	3	5
Longevity	4	2	5	5
Ease of use	3	2	4	4
Size	3	3	2	2
TOTAL	**19**	**16**	**24**	**30**

Regarding the price, PXI and VXI are powerful industrial technologies that could be a little bit expensive for be using in a remote lab because likely, not all the benefits and capacities of the mainframes are exploited. GPIB is the most common technology used instead of PXI or VXI, but a part of the limitations regarding to the distance, speed transmission and number of instruments controlled by one controller, GPIB requires specific cards and cables that could increase the price of the lab. However, LXI as it is based on LAN interface, the cost of including this interface in the equipment is not significant regard the price of entire instrument.

Furthermore since the flexibility point of view, as LXI makes use of the IVI standard, it allows that the same control software can be used to control instruments from different manufacturers without having to make changes into the control code, only in the system configuration. This presents a great advantage, because the same control algorithm can be used in different laboratories, and the update of the lab with new equipment is quick and easy.

IV. AVAILABLE SOFTWARE TECHNOLOGIES TO INSTRUMENTS AND EXPERIMENTS CONTROL

Once the hardware technology used to connect the server with the instruments is selected, we have to pay attention to the technologies that will allow control algorithms to manage the instruments and experiments. The programs that run on the server and that are developed to control the instruments can be coded by a large numbers of desktop technologies: MatLab, LabVIEW, C/C ++, AgilentVEE, .NET, Java, etc. All of them include functions and applications that make easy facilitate the control of the instruments, so the selection of one of them depends on the experience of the developed using these programming environments.

But in order to send commands to the instruments and read the responses, it is necessary to have the driver of the instrument. The driver is the set of software routines that controls a programmable instrument. Each of these routines corresponds to a set up action, a reading/writing data performance or a measurement on the instrument. In this way, using the drivers, the control of the instruments is simplified and it eliminates the need of learning the programming protocol of each instrument.

In 1993, the VXI Plug & play Systems Alliance created a set of instruments specifications called VXIPlug & play drivers. Unlike SCPI these drivers do not establish rules to control specific instruments, but they specify common aspects in an instrument driver. That is, making use of this type of drivers, it is not necessary to send a string of

ASCII text to the instrument, as it is done in SCPI, but there are a number of routines that can be called from the control algorithm. The aim of these routines is to abstract the programmer from the instrument own commands [19].

With this type of drivers, one of the main contributions of the Alliance was the development of the Virtual Instrument System Architecture (VISA). It consists on a software architecture that makes easy the communication between the control application and the instrument. This architecture provides the services and functionalities of the instruments' drivers. Thus, this standard includes a set of specifications for each of the available I/O interfaces such as GPIB, PXI, VXI, or TCP (using VXI-11). In this way, the communications interfaces used to link the control server and the instruments are transparent for the developer.

However, the VXIplug & play drivers don't provide a common programming interface. This means that the algorithm that controls an Agilent multimeter is different to that would be used to control one manufactured by National Instruments. Thus, due to the existing inconsistency when the control algorithms are carried out, programmers must spend much time learning the specific operation of each driver.

To overcome this drawback and provide a common reference to all the instruments that could be used by developers, in 1998 the Interchangeable Virtual Instruments Foundation (IVI) was established, in which are involved the most important manufacturers of test and measurement instruments. The aim of this foundation is to promote a set of specifications to program the instruments in order to enhance the equipments performance, for reducing the development and maintenance cost and for simplifying the exchange and updating of the instruments.

Thus, using IVI drivers, the same control algorithm can be used to control a multimeter regardless of the manufacturer. This is the main motivation of the development of the IVI drivers, since the specification of a driver defines an open architecture for a set of classes of instruments and a set of common software components known as IVI Shared Components. These components provide a range of common services, such as the administration of the instruments by the control system, to both the drivers and drivers' users [20].

The architecture defined by the standard VISA is also used by the IVI drivers, and as we have referenced before, LXI instruments have to implement this type of driver. In this way, control algorithms can use IVI drivers to control the LXI instruments, through the input/output functionalities provided by VISA, in particular by the VXI-11 protocol, because this protocol is used in Ethernet networks as an identification method for the instruments connected to a network. These relations between the components are displayed at Figure 2.

Once we have analyzed both types of drivers, VXIPlug&play and IVI, we have decided to use IVI drivers to develop our control methodology due to the following reasons:

- **Consistency**: all the IVI drivers are designed using a common software structure to control the instruments. This structure allows saving time when a new control software is developed.
- **Ease of use**: these drivers provide an easy, intuitive and fast access to the functions of each instrument that can be accessed from different software technologies.

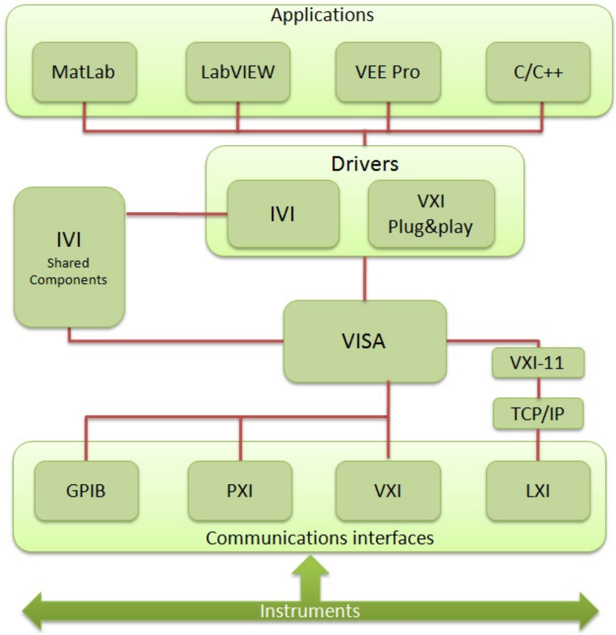

Figure 2. Software and hardware components to instruments control

- **Quality**: as IVI drivers are divided in different classes (multimeters, oscilloscopes, function generators, power supplies, RF generators, switching matrix, spectrum analyzers and power meters), it ensures that the set of instructions defined for each class is valid for all instruments regardless of the manufacturer.
- **Simulation**: IVI drivers allow that the developed control algorithm could be tested without the need to have the real instrument. That is, these drivers can simulate the behavior of the equipment
- **Data checking**: IVI drivers check the values that are sent to the device in order to verify that the configuration parameters are valid.
- **Status checking**: IVI drivers check the state of the instruments during the commands execution and operation. This characteristic allows deleting duplicate commands that could slow down the device.
- **Encourage instruments exchange**: IVI drivers allow an instrument to be replacement by another of the same class without having to recode the control algorithm. It reduces significantly the integration time in new systems or updates in existing ones.

V. PROPOSED CONTROL METHODOLOGY

As we can see at Figure 3, the proposed solution to control the hardware of the remote laboratory is based on three-level architecture. At the first level are placed the servers on which the proposed control methodology runs. This level is based on the architecture defined at the VISIR project [21], but in this case the Equipments and Experiments Server has been defined again. This recodification is due to the requirements defined by the new control algorithm. The Measurement Server has been also adapted for this new control logic.

Second level provides the communications interface. As we have analyzed before, LXI is the selected communication system to link the instruments with the Equipments and Experiments Server. Using LXI, an instruments network has been created in the lab. Each instrument has an IP address and is connected through a hub, to the equipment server placed at first level. Obviously the physical equipment in which both servers (equipment and measurement server) run is also connected to the instruments network.

In the third level we have located the hardware which contains the components or prebuilt circuits that will be part of the available experiments into the laboratory. In this hardware the instruments are also connected in order to carry out the tests and measures required for the practical exercises. This hardware is in the remote laboratory as the breadboard of prototypes that the pupil uses in traditional laboratory to build and test their circuits.

The reason why we have raised this architecture in three levels is to allow any developer to deploy a remote laboratory based on this methodology, but using the technologies that he wants to or he has available at all times. Thus, we see below as in the first level, the equipment and experiments server is developed in LabVIEW, but following the concepts proposed in this research work based on the use of IVI drivers, this server can be developed in any other programming technology and the final outs should be the same.

The same applies to the second level, because in this case we have used a network standard based on Ethernet, but if the instruments implement any other physical transmission interface on which to deploy the standard LXI (used in this solution), the operating result of the proposed solution, is exactly the same.

In the case of the third level hardware, it is a hardware whose function is only to host the components and circuits of the experiment, so it does not contain control logic and it could be built in any other way to rise in this solution. To build these boards, only certain aspects regarding the selected matrix must be taken into consideration [10].

A. *Selected software control methodology*

The control methodology is based on two servers:

Measurement server: on one hand, it is in charge of receiving the requests sent by the user through he web client and check that they can be executed. This validation is done both for instruments and the circuits that the user wants to build in using the switching matrix.

- **Equipments and experiments server**: Its main function is to execute commands processed by the measurement server and perform measures on the available instruments in the laboratory. In other words, this server is responsible of the control of the instruments and experiments, so the control methodology that we propose will be executed in this server.

In the proposed solution, the role of the measurement server is close to the functions of the server developed in the VISIR project. In the server built in this solution, several modifications have done to interoperate with the Equipments and Experiments Server that has been completely redefined.

The structure of this server (Figure 4) has been developed to allow future updates with new instruments and devices. Moreover, each developer can select the instruments available in his lab in an easy way. That is, not all the defined instruments in this methodology have to been used in all the scenarios.

The great novelty introduced in this method of control, is that all instruments are controlled and monitored using only IVI drivers. These drivers allow that only by updating certain parameters of the configuration of the server, the instruments can be replaced by others. The only one restriction is that these new instruments have to be controlled by IVI drivers.

To configure the instruments, first, the instrument IVI driver must be installed and the IVISharedComponent must be setup. When this package is setup, the file IVIConfigurationStore.xml is saved. This file contents all the updates and registers about the IVI drivers that are setup in the system. This XML file configures the relationship between the IVI driver of each instrument and its I/O reference. It also saves information regarding with the instruments' models and drivers so that just modifying the reference to the model, the instrument can be controlled without updating the code. The structure of the IVIConfigStore is shown at Figure 5.

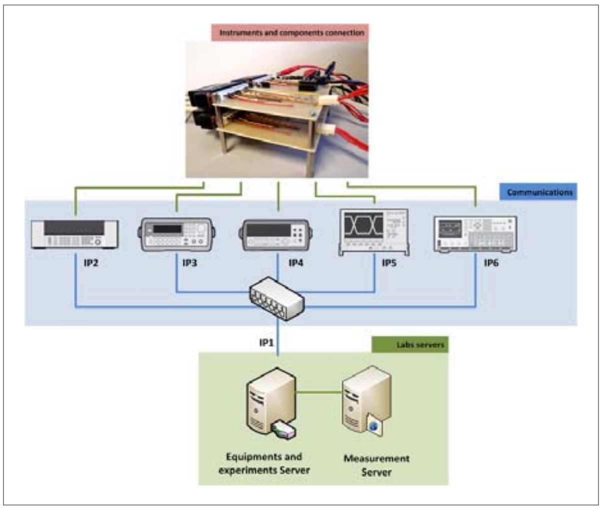

Figure 3. Proposed physical architecture

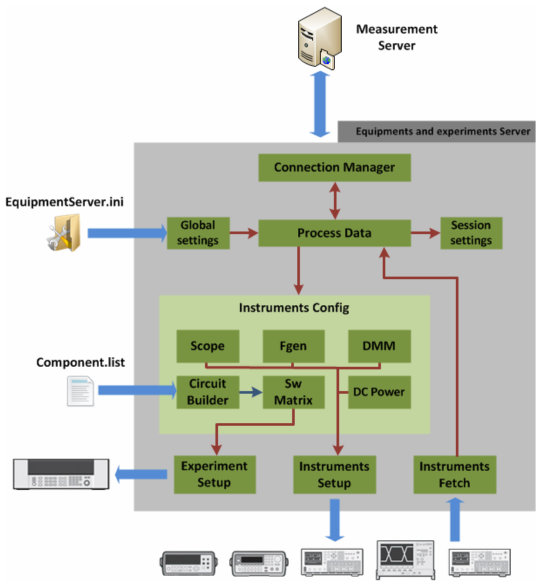

Figure 4. Equipments and Experiments Server architecture

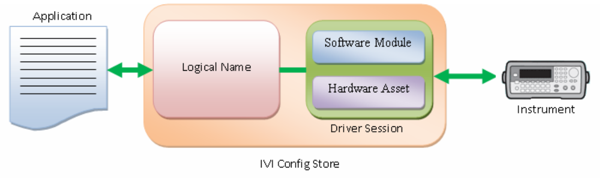

Figure 5. IVIConfig Store configuration

Through this configuration system, when we want to replace one instrument with other that complains the IVI standard, we only have to perform three simple and fast steps: a) update the 'Software Module' installing the IVI driver of the new instrument; b) update the IP address assigned in the 'Hardware Asset', c) update the description that relates both modules in the 'Driver Session'.

In this way, using generic Logical Names such as 'MyFgen' that refers to a Driver Session with a generic name like 'FunctionGenerator', the relationship Logical Name – Driver Session can be maintained unaltered during the update of the instruments of the laboratory. Thus, in the control algorithm, if the Logical Name is used when the function generator is referenced, we need not modify any part of the code although we have changed the physical instrument in the lab. This design allows the developer to use the appropriate instruments according to the lab requirements and the available resources.

In order to perform this update and configuration process, there are different tools that makes it simply: the Measurement & Automation Explorer (MAX) of National Instrument is what we have used in our methodology, although you can also find similar applications to configure applications developed in Matlab (Test & Measurement Tool) in Agilent VEE (Instrument Manager) or if the application is developed in C++, there are specific routines ready to use.

VI. RESULTS

During this year 2011 we have obtained a stable version of the control methodology that it will be tested with students during the next academic year. By the moment, all the circuits and experiments that we had implemented in the original VISIR platform have been deployed in this new system, and the performance of the lab is correct. So we can say that the proposed control methodology is available for using in a remote lab and can replace the hardware architecture proposed in the VISIR project.

Since the deployment point of view, this method makes easy the addition of new components and experiments to the lab, because the restrictions and operation of the switching matrix [10] is easier than the switching matrix defined in the VISIR project.

One of the drawbacks of the proposed solution could be the time response of the switching matrix, but we have to test the switching time in order to evaluate its performance when several students will be using it at the same time.

VII. CONCLUSIONS

In a remote lab implementation, a previous analysis is required in order to choose the appropriate software and hardware technologies, because this selection determinates the performance of the lab and the available future updates or maintained actions.

Thus, if we can develop a software independent from the hardware of the lab, we could obtain labs easily updated without code modifications. This update would be done from the instruments or the experiments point of view.

Thanks to the advantages of LXI and IVI, we can deploy a distributed test and measurement lab. And, although we have used this methodology for a learning application, the same concept and basic ideas can be used in a industrial application.

REFERENCES

[1] García-Zubia, J., López-de-Ipiña, D., Orduña, P., Hernández, U., Trueba, I. "Evolution of the WebLab at the University of Deusto" *in Proceedings of 6th European Workshop on Microelectronics Education (EWME)*, Stockholm (Sweden). Jun. 2006..

[2] García-Zubia, J., Orduña, P., Angulo, I., Irurzun, J., Hernández, U. "Towards a Distributed Architecture for Remote Laboratories" *International Journal of Online Engineering (iJOE)*, Vol.4, Special Issue REV 2008, pp. 11-14, 2008

[3] Garcia-Zubia, J., del Moral, A. "Suitability and implementation of a WebLab in engineering" *in Proceedings of 10th IEEE Conference on Emerging Technologies and Factory Automation* (ETFA), pp. 56-64, Catania, Sept. 2005

[4] Gustavsson, I., Zackrisson, J., Håkansson, L., Claesson,I., Lagö. T.L., "The VISIR project – an Open Source Software Initiative for Distributed Online Laboratories", *in Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, Porto, Portugal, June 25 – 27, 2007.

[5] Gustavsson, I. et al., "An Instructional Electronics Laboratory Opened for Remote Operation and Control", Proceedings of the ICEE 2006 Conference, San Juan, Puerto Rico, July 23 - 28, 2006.

[6] Gustavsson, I. "A Traditional Electronics Laboratory with Internet Access", Proceedings of the International Conference on Networked e-learning for European Universities, Granada, Spain, November 23 – 25, 2003.

[7] Gustavsson, I., Zackrisson,J., Håkansson, L. "A new concept for distributed laboratories based on open source technologies", in Advances on remote laboratories and e-learning experiences, Ed. by L. Gomes and J. Garcia Zubia, University of Deusto, Bilbao, 2007, pp. 247-267.

[8] Garcia-Zubia, J., Orduna, P., Lopez-de-Ipina,D., Alves, G.R. "Addressing Software Impact in the Design of Remote Laboratories" IEEE Transactions on Industrial Electronics, Vo. 56 Issue 12, pp. 4757-4767, ISSN: 0278-0046. December 2009

[9] Gustavsson, I., Olsson, T., Åkesson, H., Zackrisson, J., Håkansson, L. "A Remote Electronics Laboratory for Physical Experiments using Virtual Breadboards", Proceedings of the 2005 ASEE Annaual Conference, Portland, USA, June 12 - 15, 2005.

[10] García Zubia, J., Hernández, U., Angulo, I., Orduña, P., Irurzun, J. "LXI technologies for remote labs: an extension of the VISIR project", in Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV), Stockholm, Sweden, June 25 – 27, 2010.

[11] PXI Express Hardware Specification Revision 0.5, PXI Systems Alliance, July 15 2005

[12] Sanchez, B., Bragos, R. "Modular workbench for in-situ and remote laboratories", in Proceedings of IEEE Instrumentation and Measurement Technology Conference (IMTC), Vol. 1-5, pp. 2137-2140, Warsaw (Poland), May 2007

[13] SCPI Technical Specification, Standard Commands for Programmable Instruments (SCPI), IVI Foundation, 1999

[14] LXI Version 1.3 Specification, LXI Standard Revision, LXI Consortium 2008

[15] Browne, J. "Networking RF Tests With LXI", MICROWAVES & RF journal, Vol. 48, Issue 5, pp.54-55 , May 2009

[16] Asumadu,J.A., Tanner, R., Fitzmaurice, J., Kelly, M., Ogunleye, H., Belter J., Song Chin Koh, "A Web-Based Electrical and Electronics Remote Wiring and Measurement Laboratory (RwmLAB) Instrument," IEEE Transaction on Instrumentation and Measure-

ment, Vol. 54, No. 1, pp. 8-44, February 2005, http://dx.doi.org/10.1109/TIM.2004.834597

[17] García-Zubía, J., López de Ipiña, D., Hernández, U., Orduña, P., Trueba, I. "WebLab-GPIB at University of Deusto" in Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV), Oporto (Portugal). Jun. 2007.

[18] Nedic, Z and Machotka, J. "Remote Laboratory Netlab for Effective Teaching of 1st Year Engineering Students" International Journal of Online Engineering (iJOE), Vol.3, No. 3.

[19] VXIplug&play Specifications, VPP-2: System Frameworks Specification, IVI Foundation 2010

[20] IVI Specification, IVI-3.1: Driver Architecture Specification, IVI Foundation 2010

[21] Zackrisson,J., Gustavsson, I., Håkansson, L. “An Overview of the VISIR Open Source Software Distribution 2007”, in Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV), Porto, Portugal, June 25 – 27, 2007.

AUTHORS

U. Hernández is with the University of Deusto in the Telecommunications Department at the Faculty of Engineering. He is developer of the research group on web-based laboratories and he is in charge of the remote labs based on instruments control. He is involved too in the deployment in University of Deusto of the VISIR project leaded by the Blekinge Institute of Technology (Ronneby, Sweeden). (e-mail: unai.hernandez@ deusto.es).

J. García-Zubia is with University of Deusto, Electronics and Automation Department. He is with the University of Deusto, he is Head of Dpt. Of Industrial Electronics, Control Engineering, and Computers Architecture of the Faculty of Engineering. He is the responsible of the remote lab at the University of Deusto (WebLab-DEUSTO: weblab.deusto.es). The WebLab-Deusto has been implemented using web 2.0 techniques (AJAX, SOAP, etc.), this approach is a novelty in Europe. Different works have been published explaining the results and the technology of this weblab and the evolution of WebLab-DEUSTO has been supported by different projects. (e-mail: zubia@ deusto.es).

This article is an enxtended version of a paper presented at the International Conference on Remote Engineering & Virtual Instrumentation (REV2011), held at Transylvania University, Brasov, Romania, June 28 - July 1, 2011. Received 27 July 2011. Published as resubmitted by the authors 26 September 2011.