A Flexible and Configurable Architecture for Automatic Control Remote Laboratories

Martin Kalúz, *Member, IEEE*, Javier García-Zubía, *Senior Member, IEEE*, Miroslav Fikar, *Member, IEEE*, and Ľuboš Čirka

Abstract—In this paper, we propose a novel approach in hardware and software architecture design for implementation of remote laboratories for automatic control. In our contribution, we show the solution with flexible connectivity at back-end, providing features of multipurpose usage with different types of experimental devices, and fully configurable client side application at front-end. The physical setup and communication principles of hardware architecture are based on two types of devices: the programmable logic controllers and industrial network routers. The user interface of client application is designed as a Web page, powered by optimized JavaScript, using the sophisticated on-the-fly content generation. To prove the suitability of the architecture, we compare it with other existing approaches of remote laboratory design. We evaluate their benefits and weaknesses, especially in terms of expense, implementation difficulty, and versatility of usage. In this paper, we also show a detailed example of remote laboratory implementation based on new architecture for thermo-optical educational system and provide three other examples of developed remote laboratories. Evaluation of remote laboratory usage and its benefits is provided to demonstrate the learning value of proposed architecture in education process.

Index Terms—Remote laboratories, PLC laboratories, architectures, industrial hardware, process control

1 Introduction

The main idea of remote laboratories is to extend the possibilities of practical experimentation in scientific and educational environments.

The hardware architectures of remote laboratories can be very flexible and many different forms and their applications are available [1], [2], [3]. As it is, among the other interesting conclusions, mentioned in [4], the majority of available existing solutions are in the form of *ad hoc* or single-purpose labs, where creators adapt the architecture design and software development to a specific purpose of lab usage.

In practice, several development systems have been created to solve this issue. The *MIT iLab*[5], *Sahara*[6] and *WebLab-Deusto*[7], [8] are Remote Laboratory Management Systems (RLMSs) that provide general purpose solution to the development of Web-labs. This solution is based on a provision of Application Programming Interfaces (APIs) for interconnection of laboratory technological apparatus with system's architecture, and for programming new or integrating existing user interfaces. Although this very popular approach provides a general solution for development, it requires advanced programming and often a complex system configuration to bring new laboratories online. Moreover, with these systems, it is more difficult to develop/integrate some classes of remote laboratories than others. For example, demonstrative laboratories that provide simple visual or numerical feedback and do not offer real scientific measurements are more easy to be developed than, for example, process control laboratories. Another approach of API-based development systems is to integrate the existing laboratories. This approach also requires programming and more advanced technical skills. In general the mentioned systems (iLab, Sahara, WebLab-Deusto) consists of four functional layers [8] (Fig. 1). L1 and L2 represent the local architecture and interconnections between the laboratory and the management system. This is the place where the API-based programming applies. Other two layers represent the wider interoperable architecture, where L3 is a federation between two or more instances of the same management system and L4 is a federation between different systems.

In this work, a new type of physical framework, Multipurpose Hardware and Software Architecture (MHSA), is proposed as the complementary layer between L1 and L2 of API-based development systems (Fig. 1). The architecture is built on industrial devices such as Programmable Logic Controllers (PLCs) and Industrial Network Routers (INRs). The physical framework is designed to remove any programming requirements for new laboratory instances, and to reduce the development procedure to a few trivial *connect & configure* steps. This feature is achieved by the novel multi-purpose concept of hardware and software architecture design. As a result of this approach, significant savings has been achieved in terms of time and effort for development of new labs. On the other hand, the physical construction of the architecture limits its usage to a specific (but wide) range of applications. In general, architecture can serve all types of PLC experiments as well as any kind of laboratory or industrial apparatus that can be connected

• *M. Kalúz, M. Fikar and Ľ. Čirka are with the Institute of Information Engineering, Automation and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Slovakia. E-mail: {martin.kaluz, miroslav.fikar, lubos.cirka}@stuba.sk.*
• *J. García-Zubía is with the Faculty of Engineering, University of Deusto, Bilbao, Spain. E-mail: zubia@deusto.es.*

Manuscript received 10 Jan. 2014; revised 24 Nov. 2014; accepted 22 Dec. 2014. Date of publication 7 Jan. 2015; date of current version 14 Sept. 2015. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TLT.2015.2389251

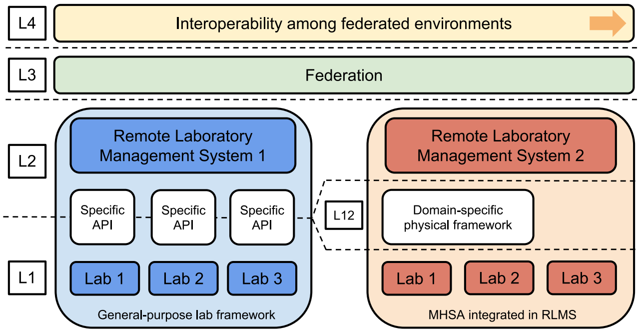


Fig. 1. MHSA as the complementary framework for remote laboratory development.

and controlled by PLCs. This makes the architecture generally suitable for automatic control remote laboratories, where real technological processes are used.

The paper is organized as follows: Section 2 provides an overview of architecture types used in related works and applications. In Section 3, a new proposed architecture for remote laboratory implementation is described, addressing both the hardware and software impact. Section 4 provides an evaluation of the proposed architecture over the experts' criteria and its comparison to other types of architectures used in related works. In Section 5 the range of architecture's applicability is discussed. A detailed implementation example of process control training remote laboratory is given in Section 6. In Section 7, a demonstration of educational benefits of proposed architecture is provided. The evaluation of students' experiences of remote experimentation is given in Section 8. Section 9 concludes the paper by summarizing the main contributions and suggesting future work.

2 Architectures in Related Works

Many different approaches to remote lab development can be found in literature. These often vary in design of hardware architecture and software used. The most frequently occurring types can be classified in the following categories.

1) *Client—server with control software—experimental device*. This commonly occurring type of architecture uses a direct connection between the PC (server) and controlled device e.g. by serial or USB port (Fig. 2—case ①), therefore the PC must be equipped with proper software for communication with the client and also for control of the experimental device. In most cases, server-side PCs contain self-made software solutions based on commonly available technologies. The main advantage of this architecture type is in its low implementation difficulty and price. In [9], the remote laboratory for control of thermo-optical plant and magnetic levitation is designed on this type of hardware architecture. The author uses the client Java application and server technology based on Java/MATLAB. Another different approach of remote lab implementation on this type of architecture is given in [10]. The authors use the LabVIEW software that is designed to handle almost all features required in remote lab operation, such as communication with experimental devices, server capabilities, data acquisition and Graphical

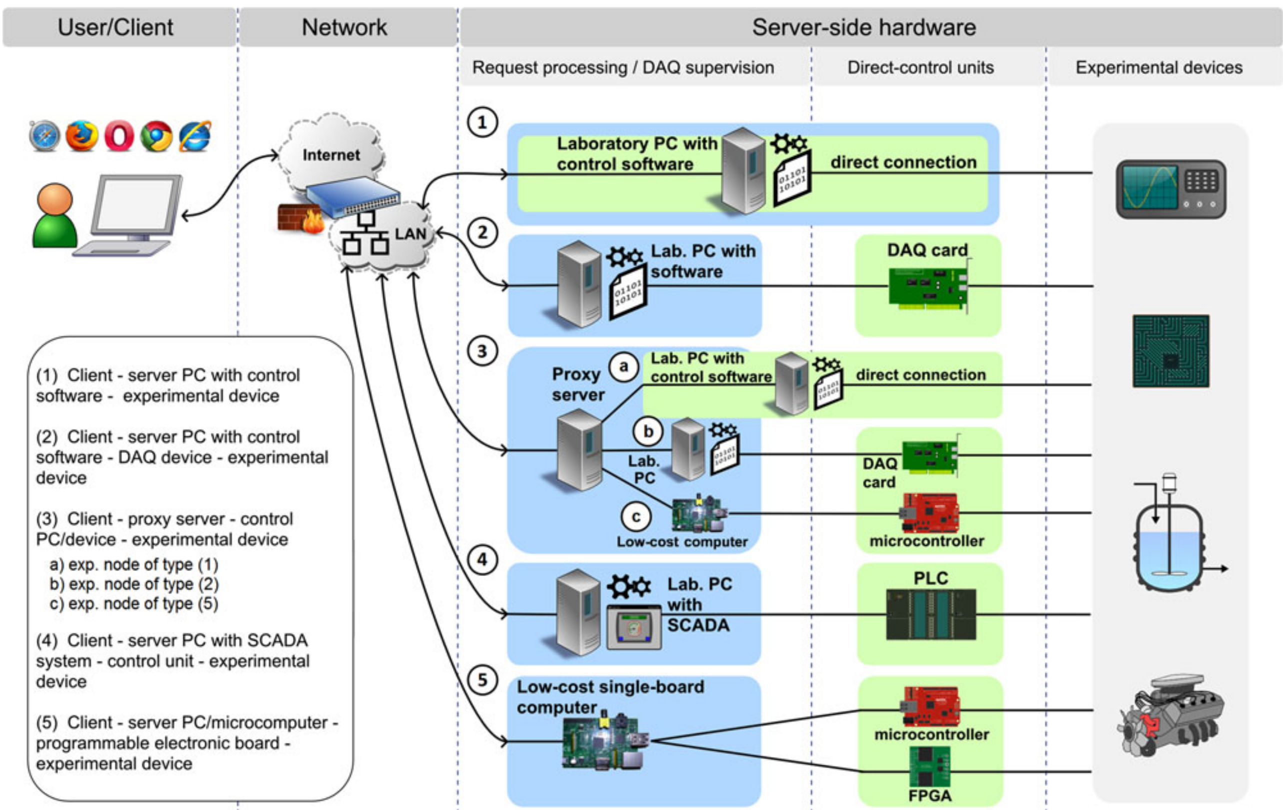


Fig. 2. Summarizing different types of architectures.

User Interface (GUI) design tools. The advantage of commercial programs like MATLAB and LabVIEW is that they provide easier implementation and time savings, but they also increase the final cost of the remote lab.

2) *Client—server with control software—Data Acquisition (DAQ) device—experimental device*. Remote experimental setup with very similar functionality as above mentioned. The only difference is that the data acquisition device (usually DAQ card) is a separate part of the architecture (Fig. 2—case ②). Architecture with a DAQ device is usually used in a situation when it is not possible to directly interconnect the server-side PC and experimental device. This type of hardware setup often reduces the requirements on signal processing software included in the control PC, but also increases the price. Control lab architectures using this type of approach are proposed in [11], [12].

3) *Client—proxy server—lab nodes(PC + control unit)—experimental devices*. This type, also known as branched architecture, provides a higher capacity of possible experimental connections than any other architecture (Fig. 2—case ③). The proxy server is used for managing the connections of different clients to different experimental nodes. Each node usually consists of a local computer (lab server) equipped with appropriate software and hardware for data acquisition, and an experimental device. The main advantage of this architecture comes from its wide inter-connectivity, where experimental nodes can be located even in different countries, while the user accesses are managed by a proxy server [13], [14], [15], [16]. Interesting branched architecture that uses wireless experiment nodes is given in [17].

4) *Client—server with supervisory control and data acquisition (SCADA) system—control unit—experimental device*. Combination of SCADA system and a control unit (PLC or other hardware controller) is one of the most frequently used industrial approaches (Fig. 2—case ④). Among the common features like data acquisition, variable control capabilities and connectivity, most SCADA systems also provide Human-Machine Interfaces (HMIs), that can be used for client side implementation of remote labs. In [18] the remote laboratory for tank level control is introduced using RSView32 SCADA system with Web-based HMI. An interesting solution combining Java-based client-server communication, Easy Java Simulations (EJS) for GUI development and SCADA-PLC control mechanism is shown in [19]. Other applications of remote laboratories based on PLCs and SCADA are shown in [20], [21]. Approaches in this category are characterized by their high cost stemming from the use of commercial software and hardware, but also by significant reduction of required implementation effort and time. Most commercial control systems contain built-in features that can be instantly used for remote control, so the developers do not need to write processing software or user interfaces.

5) *Client—server/microcomputer—programmable electronic board—experimental device*. The very popular approaches in recent years, also often labeled in literature as *low-cost*, are based on programmable boards like Field Programmable Gate Array (FPGA), Arduino micro-controllers, and cheap alternatives to standard computers (e.g. single-board computers based on ARM architecture like Raspberry Pi, Pandaboard and others) (Fig. 2—case ⑤). Due to the wide spectrum of various hardware components, this category cannot be generalized as a specific type of architecture and it differs from other types only in the use of cheap components. This approach can be described as the opposite alternative to category 4, especially from the pricing and implementation point of view. While the purchase price of hardware is incomparably lower than the price of industrial control systems, the implementation difficulty and required effort for software development rises rapidly, because developers must work with raw components. An interesting example of low-cost remote laboratory based on microcomputer and FPGA board is shown in [22]. The authors of papers [23], [24], [25] show similar approaches, but instead of a microcomputer, they use a standard PC server to provide communication between user and experimental devices.

3 Proposed Architecture

In this paper, we propose a Multipurpose Hardware and Software Architecture (MHSA)—a domain-specific physical framework for remote laboratory development and implementation. MHSA consists of hardware architecture, based on industrial aimed devices, that allows developer to connect various types of experiments into its physical setup, limited only by connectivity capabilities of experimental nodes, and client side Web based software that brings high versatility for both the final implementation and usage. This approach is physically opened at the back-end and configurable in the front-end. This allows developers to implement new instances of remote labs with minimal effort in a short time.

Comparing with other works available in literature, MHSA is novel in the different concept of development and implementation principle. Other approaches (and not only those aimed on control systems) use two main concepts: the creation of appropriate architecture for specific device, or a set of devices, creating or adapting operational software directly for such purpose (ad hoc labs); or creation of more versatile architecture that provides APIs for development, but still requires software development skills (RLMSs). The MHSA has been from the beginning developed as architecture for implementation of control system laboratories with no consideration on usage of specific device, reducing the back-end just to a set of signals. The main problem was that this concept needed client-side implementation that could adapt to different scenarios that could occur by connecting various types of experiments to the architecture. To solve this issue, we have developed fully configurable client-side application that does not require any additional programming in order to adapt its appearance or functionality to specific type

TABLE 1
Comparison of MHSA and Other Types of Architectures

Type of architecture	Cost (€)$^{(A)}$	Requirements on hardware implementation	Requirements on software development	Performance$^{(B)}$	Flexibility$^{(C)}$	Extensibility/ Connectivity$^{(D)}$
1*	hundreds	low	average	high	low	low
2*	hundreds—several thousand	low	average	high	average	average
3a*	hundreds—several thousand$^{(1)}$	average$^{(2)}$	average$^{(3)}$	high	low	average
3b*	thousands and more$^{(1)}$	average$^{(2)}$	average$^{(3)}$	high	average	average—high
3c*	hundreds$^{(1)}$	average—high$^{(2)}$	high$^{(3)}$	low—average$^{(4)}$	low	average
4*	thousands and more	average	low	high	high	high
5*	tens	average	high	low	low	average
MHSA	several thousand	low	low	high	multi-purpose$^{(5)}$	high

**: Architectures listed in Section 2.*
A: Costs express approximate expenses of hardware components and software.
B: Computation power of hardware used in architecture.
C: Flexibility shows whether the architecture is opened or closed (e.g if integration of new experiment requires significant changes in SW/HW)
D: How many nodes can managed by server-side computer and how many experiments can be connected to one node.
1: Cost is stated only for one branch of architecture including proxy server. Overall price depends on number of nodes.
2: Depends on number of components used in branched architecture. Higher number of hardware nodes results into higher implementation difficulty.
3: The requirement on software development for physical systems in branched architectures can be significantly reduced by using open-source technologies. Some of development and management systems provide direct APIs and device aimed software (iLab, LabShare, WebLab-Deusto).
4: If low cost device is used as the control unit (performs advanced computations), it may become a limiting factor for overall architecture's performance.
5: Multi-pupose architecture is opened at both, the back-end (various types of exp. devices can be connected) and front-end (client is fully configurable for all situations at back-end, e.g. use of different types of exp. devices).

of experiment. It also provides user interface that is automatically generated from the configuration file. This principle allow educators to make fast and flexible changes in remote laboratory instances even without necessity of shutting them down. Using the MHSA, developers and even students can comfortably implement new experiments to remote laboratories within several hours/days (see Section 7), contrary to other approaches where development may take a considerable period of time (see the discussion in Table 1).

Some preliminary concepts and applications related to this work have been published in [26] where an *ad hoc* laboratory based on INR and PLC is described. The problems with reproducibility of this solution have led us to idea of multi-purpose architecture. In [27], the first application of remote laboratory based on pre-final version of MHSA is shown.

3.1 Hardware and Software

The hardware architecture consists of two main control devices: the INR and PLC. Contrary to traditional types of architectures, MHSA replaces the server computer with an INR, that is primarily designed to supervise the industrial controllers connected in its local network. The INR represents the supervision layer of operational setup (Fig. 3).

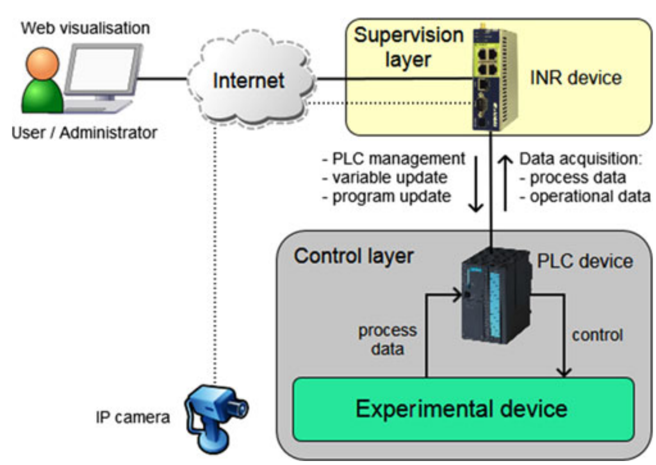


Fig. 3. Operational setup of remote laboratory architecture.

The lower operational layer consist of nodes of PLCs and experimental devices. Depending on the connectivity of specific a PLC used in architecture, provided by PLC's I/O modules, and the number of signals required for controlling the experimental device, one PLC can handle several experiments in one laboratory node. The extensional capacity of this kind of architecture is shown in Fig. 4.

The industrial router used in MHSA is eWON4005CD [28], that can be described as a coupler for industrial controllers. The main features of INR are:

- Industrial protocol translation
- Direct access to PLC program variables (read/update)
- PLC program update
- Process/operational data acquisition from PLC
- Server-side script runtime environment
- Data and event logging
- Alarms

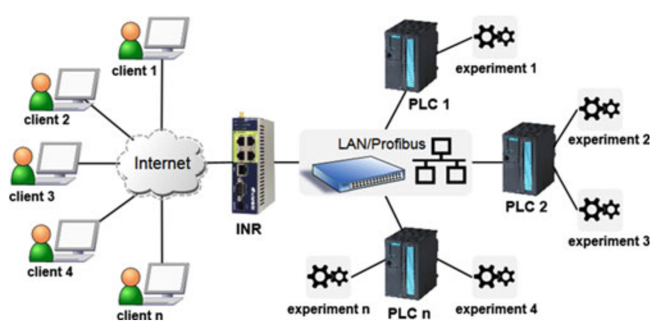


Fig. 4. Embranchment capability of hardware architecture.

- Operational architecture management
- FTP server
- Web server.

The INR device in the supervision layer communicates with all experimental nodes in the local area network by their specific industrial protocols. To simplify the communication mechanism, the INR provides the feature of associated internal variables called tags. Each tag is addressed to the specific memory area of the PLC and allows the user or administrator of the INR to directly read and update its memory variables. Tags are distributed through all program parts of the INR such as Web management screens, scripting environment and Web server services. Depending on the type of signal processing, the PLC can perform different functions in operational architecture. It can be used as non-interfering communication channel, signal processor, unit converter, and control processing environment.

The client side implementation of software architecture is based on cross-browser compatible Asynchronous JavaScript and XML (AJAX) Web application developed on Google Web Toolkit (GWT) SDK.[1] The unique point of GWT is that it provides front-end to back-end development methodology based on Java logic. The developer is not limited by lack of JavaScript standardization and does not even need to write a single line of JavaScript code during the programming phase. GWT provides a powerful compiler that uses Java code as the input source and produces JavaScript permutations of same logic optimized for all requested Web browsers. The main benefit of this approach is that developers do not have to deal with cross-browser compatibility issues, because the framework will perform this task automatically.

The MHSA provides pre-implemented security features for both communication and operation. Sensitive process data transferred between INR and PLCs are protected by specific industrial protocols. The protocol is automatically chosen by INR and it depends on particular PLC brand used in architecture. The supervisor of laboratories can use a set of alarms provided by INR. Alarms can be configured to invoke e-mail notifications and inform the supervisor about hardware/communication failures or abnormalities in process data measured in laboratory.

3.2 Client Application

MHSA provides a universal client *ControlApp*, which is designed as a JavaScript powered HTML based web application with semi-dynamic Document Object Model (DOM). The main functionality and graphical layout of the client application is defined in a separated configuration file, which contains the list of INR's referenced input signals, output signals, internal variables, configuration of graphs, video streams, main communication setup, etc.

In the comparison with other Web-based laboratories, the *ControlApp* provides a new concept of GUI construction—on-fly content generation of remote lab interface. It has been designed for specific needs of MHSA. GUI is automatically generated during the load of application, reflecting the specific configuration for a particular laboratory.

1. Available online at: http://gwtproject.org/

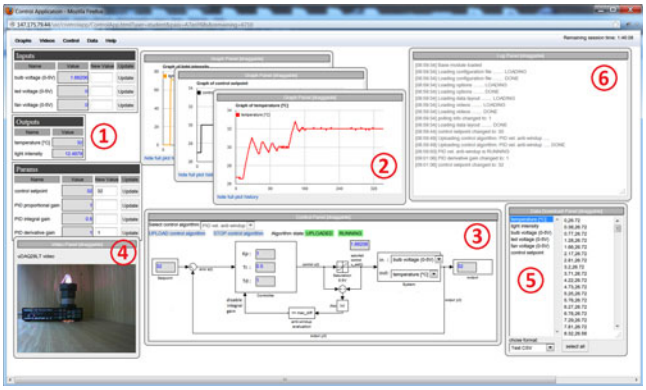


Fig. 5. Layout of graphical components in client application.

The graphical layout of the client application provides users with the following features (Fig. 5):

- table of output signals (view only) ①
- tables of input signals and additional variables (view/update) ①
- main toolbar
- graphs of signals ②
- control algorithm window ③
- custom video streams ④
- data download window ⑤
- logging history window ⑥.

Because all components of client application are defined in configuration files, the changes of client application do not require interference with the source code. This approach significantly reduces time and required effort. Control algorithms are defined in another set of configuration files and can also be easily edited. The control logic itself is defined in BASIC language that is used by the INR extensional program runtime. To add a new control algorithm in a remote laboratory, the administrator needs to write an Extensible Markup Language (XML) file with a predefined structure given by an available template. The file contains definition of graphical layout for control scheme and definition of control logic. Several different examples of the labs built on MHSA will be given in Sections 6 and 7.

4 MHSA Evaluation and Comparison with Other Types of Architectures

The MHSA provides several features that are contributive to remote laboratory development. The most significant are: the possibility to connect various types of automatic control experiments without the need of changes in architecture or source code; embranchment capacity of architecture (more than one experimental node behind the INR); significant time savings in implementation process; and two control layers that allow the use of advanced process control methods such as adaptive control, model predictive control, etc.

The main advantages/disadvantages of MHSA are summarized in the following lists.

Advantages of MHSA:

- INR substitutes PC in architecture
- reduced time and effort of implementation

- no server-side programming required (services already implemented in INR)
- allows direct use of AJAX to industrial hardware
- fully configurable cross-browser client application
- supports most PLC brands on the market (also industrial protocols)
- two hardware layers for control algorithms (INR, PLC)
- SCADA system not required
- one INR can manage up to 16 PLCs.

Disadvantages of MHSA:

- architecture is bound to specific type of INR (eWON COSY 141, 2005CD/4005CD, 2101CD/4101CD, 2104/4104)
- laboratory node must contain a PLC type supported by the INR
- limited INR memory (14 MB for Web server, 11 MB internal/script memory)
- limited PLC operational memory (depends on type)
- limited possibilities of scripting in the INR (BASIC language—only one script at a time)
- communication between client application and the INR is limited to the use of standard Hypertext Transfer Protocol (HTTP) request-response model.

Table 1 compares architectures from Section 2 and MHSA, evaluating costs, requirements on: hardware implementation/software development, performance, flexibility, and extensibility/connectivity.

In [29], important experts' criteria for evaluation of remote laboratories have been established. The authors asked eight independent experts from different universities to assess the relevance of 10 main lab characteristics in order to address their importance for development and usage.

To properly evaluate the suitability of MHSA we adopted these criteria and analyzed their satisfaction according to their importance addressed in [29].

The characteristics are following.

- *Cross-platform*. Remote laboratory (RL) can be accessed from any operating system (MS Windows, Linux, Mac OS, ...).
- *Security*. RL uses HTTP and does not need permissions on the firewalls.
- *Web browsers*. RL can be used from all Web browsers.
- *Intrusiveness*. The user does not give permissions to the client application of RL: hard-disk access, execution of native code, etc.
- *Interaction*. RL needs to implement the maximum of interaction with the user.
- *Installation*. RL runs without any previous installation in the client side: plug-in, JVM, Flash Player, etc.
- *Devices*. RL can be accessed by all the devices: PC, PDA, mobile phone, etc.
- *Bandwidth*. RL needs the maximum bandwidth efficiency.
- *Audio*. RL needs the maximum of audio and video power.
- *Power*. RL is very complex and needs a powerful tool to be implemented.

The evaluation, along with the point-to-point explanation, is provided in Table 2.

TABLE 2
Evaluation of MHSA over the Experts' Criteria Established in [29]

Characteristic	Importance (1-highest)	Level of satisfaction$^{(1)}$
Cross-platform	1	$5^{(A)}$
Security	2	$5^{(B)}$
Web browsers	3	$5^{(C)}$
Intrusiveness	4	$5^{(D)}$
Interaction	5	$4^{(E)}$
Installation	6	$5^{(F)}$
Devices	7	$3^{(G)}$
Bandwidth	8	$3^{(H)}$
Audio & Video	9	$2^{(I)}$
Power	10	$3^{(J)}$

1: MHSA evaluation (5—best, 1—worst)
A: Client application runs in Web browser on any platform.
B: The communication is based on HTTP.
C: Client application built on GWT is fully supported by all major Web browsers.
D: HTML and JavaScript are non-intrusive.
E: Client application provides a set of easy-to-use controls, interactive charts, customizable component layout, and online interaction.
F: User does not require any plugin or specialized software to access and use laboratory.
G: The technologies used in client application allow users to access laboratories from mobile devices. However the user interface is designed for comfortable usage mostly on desktop computers, notebooks and big screen devices.
H: INR supports maximum bandwidth to local network at 100 mbps. Bandwidth required by client application proportionally grows with number of video streams attached to lab.
I: Client application provides streamed or pseudo-streamed video from controlled experiment. No audio or augment reality component are incorporated.
J: Industrial systems provide less computational power than PCs, but they run only software necessary for RL operation.

5 APPLICATION SCOPE

In this section we discuss and pick up those applications for which the MHSA is generally suitable and also those where architecture can be used with some limitations.

The MHSA is generally suitable for implementation of remote laboratories for technological processes that uses sensors and actuators operated by standard electrical signals. These include processes of chemical technology such as chemical reactors, distillation columns, hydraulic systems (example of a three tank system is given in Section 7), thermal transfer processes (example of an air-flow heat exchanger is given in Section 7), and other types of laboratory training devices (example of laboratory implementation with thermo-optical device is provided in Section 6.1).

Even though the MHSA is based on devices aimed at industrial usage and with some additional operation-based security modifications, it can be also used for smaller industrial applications, the architecture itself is mainly designed for implementation of remote laboratories in academic scale.

Other types of experiments that can be easily connected to the architecture are those based on electro-mechanical actuators (example of DC motor is given in Section 7). This extends the usability of the architecture also to some specific applications in the field of robotics, mostly for robotic arms and other types of manipulators. Although we do not show the example of application in robotics, available works in literature provide some examples of PLC-based control of robotic devices [30], [31]. As MHSA is PLC-based architecture it is capable to serve these kinds of devices as well.

On the other hand, PLCs are unsuitable for other robotic applications, especially those operated by structured instructions and those operated wirelessly.

6 Implementation of Remote Lab: Example

To show the versatility of MHSA, the practical case of remote laboratory implementation is given below.

6.1 Experimental Device

For demonstration purposes we have chosen the training experimental plant uDAQ28/LT [32] as the remotely controlled device. The selected device is a multi-input multi-output thermo-optical plant designed for educational purposes in process control related fields. It is suitable for signal processing, experimental system identification, data acquisition and various types of control tasks. The plant provides three inputs (voltage for bulb, fan, and light diode) and eight outputs which can be measured (actual and filtered temperature inside tube, reference environment temperature, actual and filtered light intensity inside tube, fan RPM and current taken by fan). Several transfer subsystems of the plant can be measured. The physical description and detailed mathematical model of plant is provided in [33].

6.2 Connecting the Device to Hardware Architecture and Signal Configuration

The plant provides analog signals of inputs and outputs in one parallel connector, where each signal is kept in separate line. For the showcase, we have chosen three inputs and two outputs to be connected to the remote laboratory. Input signals for bulb voltage, LED voltage and fan voltage, and output signals for temperature and light intensity were physically connected to the PLC's (Siemens S7-300) I/O module. Inside the PLC's program logic, the simple unit conversion and signal standardization were created, and results were addressed to the specific memory areas of the PLC, from there they can be collected by the INR. In internal memory of the INR, associated tag variables for each signal were created and linked to their addresses in the PLC. The additional variables used in the remote lab, such as parameters for controller implementation and control setpoint, were also created in the INR's logic as tags.

All physical devices present in hardware architecture of this example are shown in Fig. 6.

6.3 Configuration of Client Application

The client application is in the form of a ready-to-use file package. Before Web application files are uploaded into the INR's web server, the configuration of the experiment has to be done. Each instance of client application contains a set of configuration files for different purposes.

The *tagXML.shtm* is a script designated to collect tag values from the INR. It is based on the principle of server active pages and its generated content is periodically called by the client through AJAX requests when the user of the remote lab is online. The content of the file is generated in XML structure and during the initial configuration, all necessary variables used in the remote lab client have to be listed in it.

Fig. 6. Devices in remote lab physical setup: ① controlled experiment (thermo-optical plant uDAQ28/LT), ② IP camera, ③ PLC Siemens S7-300, ④ INR device eWON4005CD.

The *clientConfig.xml* is the main configuration file that defines the following settings.

1) *Definition of signals and variables.* For each signal linked to the INR, the configuration file contains its tag name, descriptive name that appears in the client application, minimum allowed, maximum allowed, and default value. This section also contains specialized variables which are processed in the background of the application, such as the state of running control algorithm, the time signal and other.
2) *Definition of graphical layout.* These settings contain definitions of graphs for each signal/variable and video frames. In this way the administrator can define formatting of graphs, their appearance and functionality. Video frames allow two types of data sources, the MJPEG stream (supported by Mozilla Firefox, Google Chrome, and Apple Safari) and JPEG snapshot simulated stream (Internet Explorer, Opera, etc.).
3) *Communication settings.* This section defines URLs of all necessary services used by client application.
4) *General usage restrictions.* The restrictions are defined for cases of unexpected user and application behavior.

Control algorithms are included in client application as another set of modular XML files. The directory tree contains one file which defines list of algorithms, paths to their sub-directories, and names of XML files with algorithm definitions. The XML list file for this example provides three types of control algorithms: simple relay controller; PID velocity controller; and the same PID controller with anti-windup connection. The list of control algorithms is parsed at the load of client application and the program logic provides the user with their selection. The XML file for each control algorithm contains general information, graphical layout definition, and two sections of control logic in BASIC language (initial, cyclic). When the user of the client application selects a specific algorithm, the program automatically loads and parses its XML file. Sections of BASIC language with representation of the control algorithm are loaded into

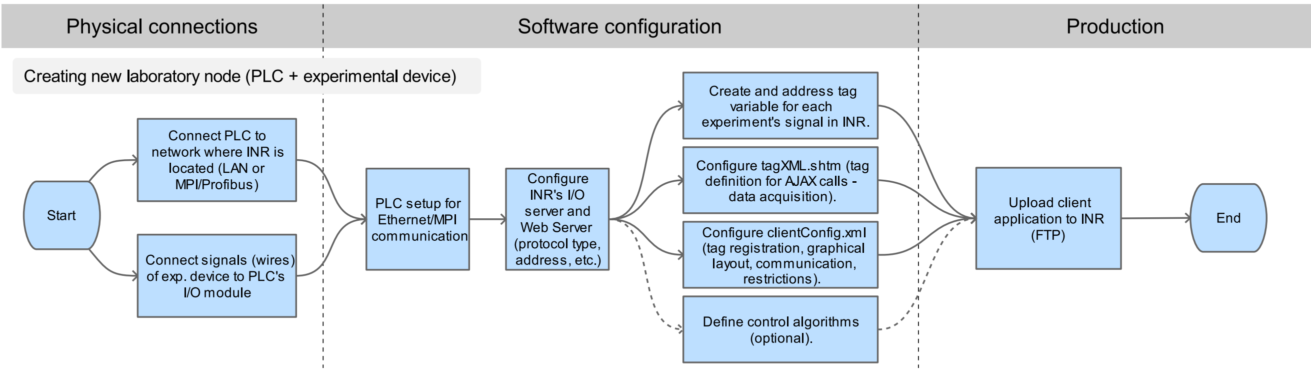


Fig. 7. Task diagram: Creating new remote laboratory node (PLC + experimental device).

client application, from where they are uploaded to INR runtime environment on user's request.

6.4 Final Implementation and Usage

The final phase of remote lab implementation includes the upload of client application and configuration files into a Web server of the INR device. For this purpose, the INR device provides an FTP server to access the directory tree of the Web server. Uploaded applications do not require any additional setting and can be used instantly.

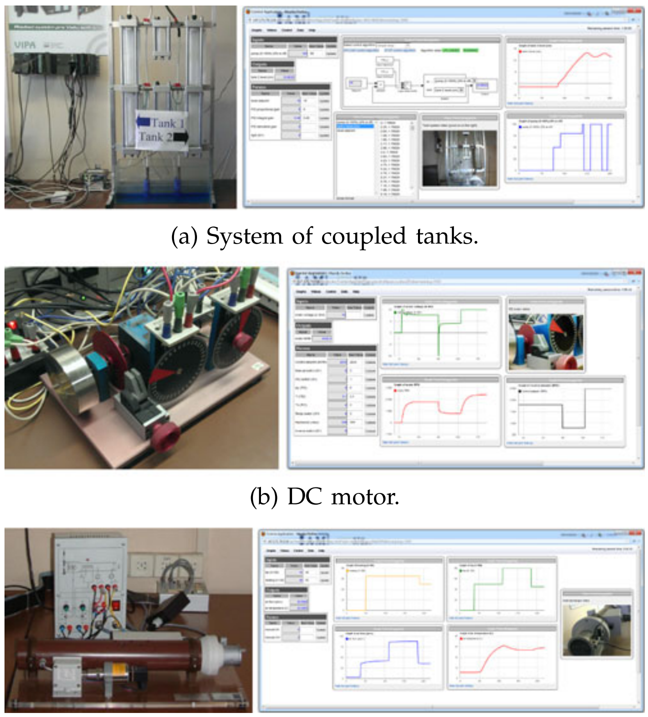
The whole procedure of remote laboratory node implementation is shown on task diagram (Fig. 7). The implementation tasks are split into three steps. The first involves the physical interconnection of individual hardware parts into the final architecture. This task must be performed by a lab developer, since a deeper knowledge of experimental device properties is required. The second step contains three configuration tasks: communication setup of PLC; configuration and communication setup of INR; and overall configuration of client application. In the implementation phase, the system administrator is responsible for these three tasks. The last step is production (upload of client application into Web server of INR) and it can be performed by system administrator or teacher with system permissions. Though these tasks are normally handled by technically skilled persons (developers and educators), in Section 7 we show that they can be easily carried out also by students that wish to participate on remote laboratory development.

All process control laboratories based on MHSA are published on the Internet and are accessible through RLMS WebLab-Deusto deployed at Slovak University of Technology (STU) in Bratislava (http://weblab.chtf.stuba.sk). Since 2012, the WebLab-Deusto directly supports MHSA-based laboratories [34]. This feature has been implemented as the result of international cooperation on remote laboratory development and federation between STU and University of Deusto in Bilbao. The main idea was to provide students of both universities with new types of laboratories and to extend the inter-institutional cooperation.

7 Experiments in Education

In the past, the educators at STU were limited by the deficiency of experimental equipment available to students. This resulted in situations where several students had to work on the same laboratory process at one time during the laboratory exercises. To solve this issue, several remote laboratories were incorporated into courses to provide students with opportunity to perform experiments 24/7.

Based on high versatility of architecture, we were able to provide students with various laboratories focused on automatic control. Overall four systems have been implemented on MHSA: thermo-optical plant (described in Section 6), hydraulic system with coupled tanks (Fig. 8a), DC motor (Fig. 8b), and air heat exchanger (Fig. 8c). The MHSA-based remote laboratories have been used in education process at Institute of Information Engineering, Automation and Mathematics for the period of two terms. Several different courses have been extended by the use of remote laboratories, namely "Process Control", "Integrated Control in Process Engineering", "System Identification", and "Process Modeling". These courses provide students with the theoretical knowledge and practice experience in automation, control systems, identification, and modeling



(a) System of coupled tanks.

(b) DC motor.

(c) Air heat exchanger.

Fig. 8. Different remote laboratories.

TABLE 3
Students' Questionnaire

Questions regarding students' experiences	
Q1	How do you grade the level of interactivity and overall technical quality of RL?
Q2	How far you have been able to technically understand the usage of RL? (Please take into account any technical/usage issues you were facing.)
Q3	How the use of RL facilitated your learning of subject?
Q4	Can you express the degree in which the RL helped you to understand the nature of experiments?
Questions regarding the usage, aimed on improvement of RL	
Q5	Have you been able to perform all subject-related tasks without technical issues? If no, please describe them.
Q6	Do you consider the current features of RL to be satisfactory? If no, please provide suggestions to improvements.
Q7	Have you been satisfied with accessibility and provided session time of RL? If no, please provide suggestions.
Question regarding the possible improvement of education process	
Q8	What ratio of remote/hands-on exercises do you prefer to perform laboratory practices in future?

of technological processes. During the first and second term of 2012/13, the students were encouraged to perform their practical tasks in remote laboratories as an extension to traditional exercises. The evaluation of students' feedback is provided in Section 8.

Master students at STU are not limited only to the use of various processes, but we also try to encourage them to participate on their creation. One of the main benefits of proposed architecture, the ease of remote laboratory implementation, has been proven by students in their participation on remote laboratory development. While two of mentioned laboratories (thermo-optical plant and hydraulic system) have been developed by educators, the DC motor and the heat exchanger lab is the result of students' work during their semestral projects. In these projects, students have learned the basic concepts of remote laboratory implementation, including instrument preparation and connection to the architecture, PLC configuration, definition of signal/unit converters and control algorithms in PLC, interconnection of signals/variables between PLC and INR, and final configuration of client application. Overall four students have been assigned into two project groups, working in pairs. These two projects were a part of master's study course "Semestral Project II".

Though the creation of fully operable remote laboratory can be a very time consuming task even for skilled developers, students were able to use MHSA and to create two new labs within several weeks in addition to their regular study duties. Moreover, these two laboratories have been deployed to education process and have been used by other students to carry out their study tasks. The main contribution for participating students was that they acquired essential knowledge and practical experience of remote laboratory development without need of advanced programming skills.

8 Evaluation

To validate the benefits of MHSA-based remote laboratories in learning, we evaluated students' experience in the terms of usage, learning perspective, and asked for suggestions how these remote laboratories could be improved both in usage and context of education. The evaluation was provided by 32 students participating in above mentioned courses. To get as authentic feedback as possible, we questioned only those students that spent at least four sessions in at least two different remote laboratories. A questionnaire (Table 3) was designed to provide basic, but targeted, evaluation of developed laboratories in two aspects. First, to examine the quality of experience provided by user interfaces. Second, to analyze the ability of remote lab to mediate the physical nature of experiment and carry out the subject-related tasks. The formulation of Q1-Q4 (rated questions) was inspired by [35], where similar types of questions were asked in order to evaluate technical aspects of lab usage. Q5-Q7 (yes/no questions) were formulated to get feedback for possible improvements of remote laboratories. As the complementary feedback to these questions, students provided written description of problems experienced during the lab sessions, their observations, and suggestions for future improvements. In the Q8, the students were asked to provide their personal preference of method for performing a laboratory exercise. This question was asked in order to adjust an appropriate proportion between traditional and technologically enhanced learning.

The overall summary of survey (Fig. 9) indicates the generally positive effect of remote experimentation in the courses. The most positive feedback was given to Q1 and Q2 as the technical impression provided by client application, incorporating the quality and usage comfort. The answers to Q3 and Q4 have shown that students gave generally favorable response to learning value provided by remote experimentation and their ability to understand the physical principles of such experiments.

Some valuable information has been collected as the additional answers to Q5-Q7. The most common technical issues the students were facing during remote sessions were associated to connection speed. Several students that used slow Internet connection have observed more significant response delays while interacting with remote experiments. Few objections were aimed on accessibility, where students had to wait for their laboratory session because the experiment has been reserved by some other student at that time. However, from the nature of remote experimentation, this issue is obvious and can be solved only by enlargement of available experiments. Two students also mentioned the need of some interactive help in client environment, to

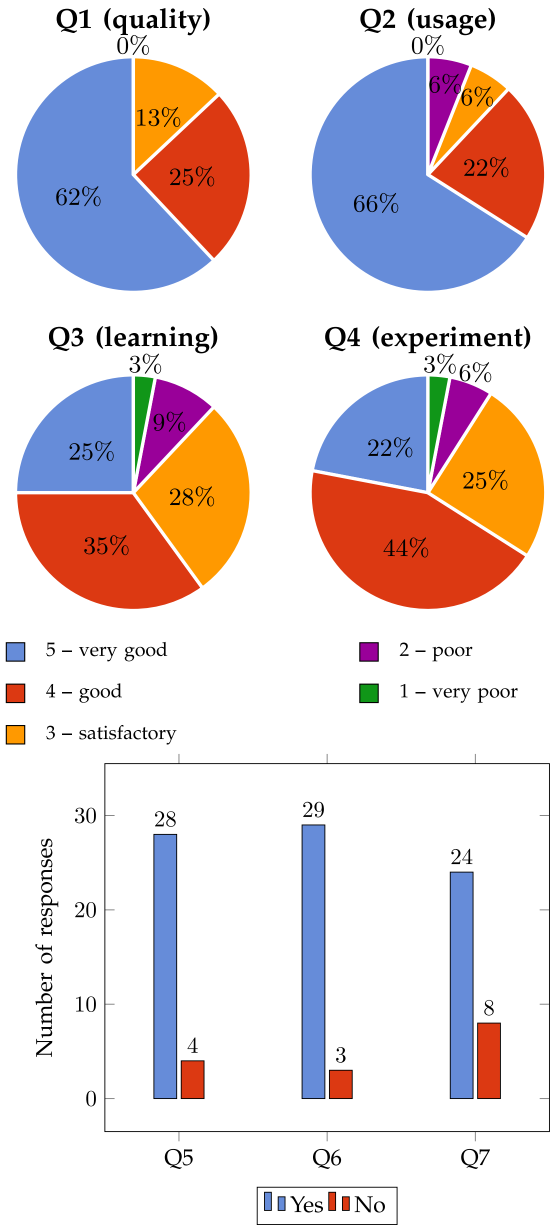


Fig. 9. Evaluation of students' experience.

enhance the features of remote laboratory for those who use it for the first time. We list some selected comments to Q5-Q7:

"The Internet laboratory is fascinating for me. I was able to measure data sets for all tasks given in Identification course in less than hour. However it took some time for me to understand how to use laboratory tools, maybe some kind of interactive help could be provided."

"The response of page drops down several times during my second session. It was much better when I closed the video from camera."

Very important suggestions have been collected by evaluating question Q8 (Fig. 10), where students provided their opinion how laboratory practices should be balanced over the remote and in-lab experimentation. The most of questioned students suggested the use of both educational approaches. On the other side, the fact that almost half of students provided the answers 1 (hands-on only) and 2

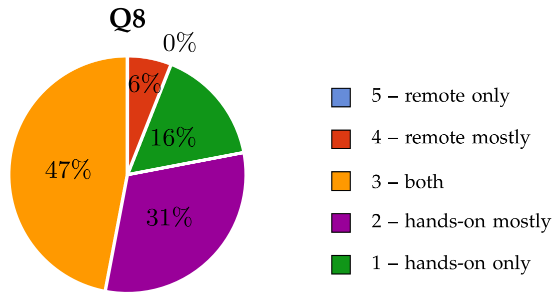


Fig. 10. Evaluation of students' suggestion in order to improve laboratory exercising.

(hands-on mostly) indicates that traditional way to perform laboratory practices still has a very important role in learning. We assume that this observation can be explained as the result of students' long-term experience with practical exercising during their studies and lack of confidence regarding the adaptation to changes of learning paradigms. This view is also supported by the results in other works. For example, a similar phenomenon was encountered in [36], where classes were performed in three modes: traditional (proximal), remote, and simulation. After the experiment, students were asked which form do they prefer for laboratory exercising. Regardless of which form the students undergone, the majority (at least 60 percent) have chosen the traditional form. This evaluation indicates the importance of appropriate distribution of learning methods between traditional and technologically enhanced.

The questionnaire also provided a space for free comments. Some of them are provided above.

"Very good tools for home exercising."

"I have tried all laboratories in Weblab and must say that they are great. A lot of control tasks could be handled there. I do not even need to go into school labs to measure my data."

"It helped me to understand the principles of automatic control and to link my theoretical knowledge with practice."

"I enjoyed the course, because I was able to test all thought concepts in labs even from home."

9 Conclusion and Future Work

In this paper we have shown a novel approach in remote laboratory architecture development and implementation, based on a combination of industrial hardware and web based operational software. This architecture represents a complementary physical framework for development of PLC-based and process control remote laboratories in much easier way than other systems which use APIs and complex configuration. The client software enables administrators of labs to create customized user interfaces and to extend them without interfering with program sources. The main limitation of MHSA is the fact that it can be used only to develop laboratories based on PLCs. However, despite this limitation, the domain of possible uses is still very wide and generally suitable for automatic control remote laboratories.

The practical cases of implementation have shown that this approach is easily reproducible for both educators and students, and its applications can be useful for different educational institutions. Several process control systems have been implemented to remote labs and used as the extension to traditional laboratory exercises. The students provided a valuable feedback in questionnaire and positively rated the deployment of such paradigm in education.

Moreover, students provided valuable suggestions to possible improvements of remote labs. We will focus on these points in our future work. Other consideration for the future work is focused on implementation of advanced process control scenarios which could reap the benefits of MHSA's layered operational structure. Application of scenarios such as online and explicit model predictive control, adaptive control and others in remote labs could be an interesting additional value to process control education. Since the participation of students in remote laboratory development has brought positive results, we plan to encourage them even more to participate on our further work.

ACKNOWLEDGMENTS

The authors acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grant 1/0053/13 and the Slovak Research and Development Agency under the project APVV-0551-11. Martin Kaluz is the corresponding author of the article.

REFERENCES

[1] L. Gomes and J. García-Zubía, eds., *Advances on Remote Laboratories and E-learning Experiences*. Bilbao, Spain: University of Deusto, 2007.

[2] J. García-Zubía and G. R. Alves, eds., *Using Remote Labs in Education: Two Little Ducks in Remote Experimentation*. Bilbao, Spain: University of Deusto, 2011.

[3] O. Dziabenko and J. García-Zubía, eds., *IT Innovative Practices in Secondary Schools: Remote Experiments*. Bilbao, Spain: University of Deusto, 2013.

[4] L. Gomes and S. Bogosyan, "Current trends in remote laboratories," *IEEE Trans. Ind. Electron.*, vol. 56, pp. 4744–4756, Dec. 2009.

[5] V. J. Harward, J. A. del Alamo, S. R. Lerman, P. H. Bailey, J. Carpenter, K. DeLong, C. Felknor, J. Hardison, B. Harrison, I. Jabbour, P. D. Long, M. Tingting, L. Naamani, J. Northridge, M. Schulz, D. Talavera, C. Varadharajan, W. Shaomin, K. Yehia, Z. Rabih, D. Zych, "The iLab shared architecture: A web services infrastructure to build communities of internet accessible laboratories," *Proc. IEEE*, vol. 96, no. 6, pp. 931–950, Jun. 2008.

[6] D. Lowe, S. Murray, L. Weber, and M. de la Villefromoy, "Labshare: Towards a national approach to laboratory sharing," in *Proc. 20th Australasian Associat. Eng. Educ. Conf.*, Univ. Adelaide, Adelaide, Australia, pp. 458–463, Dec. 2009.

[7] P. Orduña, J. Irurzun, L. Rodriguez-Gil, J. García-Zubia, F. Gazzola, and D. Lopez-de Ipina, "Adding new features to new and existing remote experiments through their integration in WebLab-Deusto," *Int. J. Online Eng.*, vol. 7, pp. 33–39, Oct. 2011.

[8] P. Orduña, *Transitive and scalable federation model for remote laboratories*, PhD Thesis, Univ. Deusto, Bilbao, Spain, may 2013.

[9] P. Bistak, "Remote laboratory server based on Java Matlab interface," in *Proc. 14th Int. Conf. Interactive Collaborative Learn., 2011*, pp. 344–347, Sep. 2011.

[10] A. Mohtar, Z. Nedic, and J. Machotka, "A remote laboratory for microelectronics fabrication," in *Proc. 38th Annu. Frontiers Educ. Conf.*, pp. S2F–7–S2F–12, Oct. 2008.

[11] A. Choudhary, S. Singh, M. Malik, A. Kumar, M. Pathak, and V. Kumar, "Virtual lab: Remote access and speed control of DC motor using Ward-Leonard system," in *Proc. IEEE Int. Conf. Technol. Enhanced Edu., 2012*, pp. 1–7, Jan. 2012.

[12] A. Chandra and C. Venugopal, "Novel design solutions for remote access, acquire and control of laboratory experiments on DC machines," *IEEE Trans. Instrum. Measurement*, vol. 61, no. 2, pp. 349–357, Feb. 2012.

[13] W. Hu, G. Liu, and H. Zhou, "Web-based 3D control laboratory for remote real time experimentation," *IEEE Trans. Ind. Electron.*, vol. 60, no. 10, pp. 4673–4682, Oct. 2013.

[14] I. Santana, M. Ferre, E. Izaguirre, R. Aracil, and L. Hernandez, "Remote laboratories for education and research purposes in automatic control systems," *IEEE Trans. Ind. Inform.*, vol. 9, no. 1, pp. 547–556, Feb. 2013.

[15] D. Lowe, S. Murray, E. Lindsay, and D. Liu, "Evolving remote laboratory architectures to leverage emerging internet technologies," *IEEE Trans. Learn. Technol.*, vol. 2, no. 4, pp. 289–294, Dec. 2009.

[16] M. Tawfik, E. Sancristobal, S. Martin, R. Gil, G. Diaz, A. Colmenar, J. Peire, M. Castro, K. Nilsson, J. Zackrisson, L. Håkansson, and I. Gustavsson, "Virtual instrument systems in reality (visir) for remote wiring and measurement of electronic circuits on breadboard," *IEEE Trans. Learn. Technol.*, vol. 6, no. 1, pp. 60–72, Jun. 2013.

[17] L. Cui, F. P. Tso, D. Yao, and W. Jia, "Wefilab: A web-based wifi laboratory platform for wireless networking education," *IEEE Trans. Learn. Technol.*, vol. 5, no. 4, pp. 291–303, Jan. 2012.

[18] G. Hongyan, "Development of a remote laboratory for process control experiments," in *Proc. Int. Conf. E-Health Networking, Digital Ecosystems Technol.*, vol. 2, pp. 87–90, Apr. 2010.

[19] E. Besada-Portas, J. A. Lopez-Orozco, L. de la Torre, and J. M. de la Cruz, "Remote control laboratory using ejs applets and twincat programmable logic controllers," *IEEE Trans. Edu.*, vol. 56, no. 2, pp. 156–164, May 2013.

[20] R. Marques, J. Rocha, S. Rafael, and J. F. Martins, "Design and implementation of a reconfigurable remote laboratory, using oscilloscope/PLC network for WWW access," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2425–2432, Jun. 2008.

[21] C. Lazar and S. Carari, "A remote-control engineering laboratory," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2368–2375, Jun. 2008.

[22] R. Costa, G. Alves, and M. Zenha-Rela, "Reconfigurable IEEE1451-FPGA based weblab infrastructure," in *Proc. 9th Int. Conf. Remote Eng. Virtual Instrum.*, pp. 1–9, Jul. 2012.

[23] R. Hashemian and T. Pearson, "A low-cost server-client methodology for remote laboratory access for hardware design," in *Proc. 39th IEEE Frontiers Edu. Conf.*, pp. 1–5, Oct. 2009.

[24] W. El Medany, "FPGA remote laboratory for hardware e-learning courses," in *Proc. IEEE Region Eighth Int. Conf. Comput. Technol. Electr. Electron. Eng.*, pp. 106–109, Jul. 2008.

[25] J. Neto, S. Paladini, C. Pereira, and R. Marcelino, "Remote educational experiment applied to electrical engineering," in *Proc. 9th Int. Conf. Remote Eng. Virtual Instrum.*, pp. 1–5, Jul. 2012.

[26] M. Kalúz, R. Halás, P. Ďurina, R. Valo, and Ľ. Čirka, "Remote laboratory based on industrial hardware," presented at the Proc. 10th Int. Sci. Tech. Conf. Process Control, Kouty nad Desnou, Czech Republic, Jun. 11–14, 2012.

[27] M. Kalúz, Ľ. Čirka, and M. Fikar, "Simplifying the implementation of remote laboratories in educational environments using industrial hardware," in *Proc. 19th Int. Conf. Process Control*, M. Fikar and M. Kvasnica, eds., Štrbské Pleso, Slovakia, pp. 522–527, Slovak Univ. Technol. Bratislava, Bratislava, Slovakia, 2013.

[28] eWON Inc., *Industrial VPN LAN Router eWON 2005CD/4005CD Product data sheet*, Nov. 2011.

[29] J. Garcia-Zubia, P. Orduna, D. Lopez-de Ipiña, and G. Alves, "Addressing software impact in the design of remote laboratories," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4757–4767, Dec. 2009.

[30] S. Stankovski, L. Tarjan, D. Skrinjar, G. Ostojic, and I. Senk, "Using a didactic manipulator in mechatronics and industrial engineering courses," *IEEE Trans. Edu.*, vol. 53, no. 4, pp. 572–579, Nov. 2010.

[31] Q. Zhong, "Design of manipulator based on plc," in *Proc. IEEE Fifth Int. Conf. Advanced Comput. Intell.*, pp. 988–992, 2012.

[32] M. Huba, P. Kurčík, and M. Kamenský, *Thermo-optical device uDAQ28/LT*, Slovak Univ. Technol., Bratislava, Slovakia, 2006.

[33] F. Jelenčiak, P. Ťapák, andM. Huba, "Mathematical Modelling and Identification of Thermal Plant," in *Proc. 17th Int. Conf. Process Control*, M. Fikar and M. Kvasnica, eds., Štrbské Pleso, Slovakia, pp. 219–225, Slovak Univ. Technol,. Bratislava, Slovakia, 2009.

[34] M. Kalúz, J. García-Zubía, P. Orduñ, M. Fikar, and Ľ. Čirka, "Sharing control laboratories by remote laboratory management system weblab-deusto," in *Proceedings of 10th IFAC Symposium on Advances in Control Education*, S. Dormido, ed., vol. 10 of *Advances in Control Education*, pp. 345–350, Univ. Sheffield, Sheffield, United Kingdom, International Federation of Automatic Control, 2013.

[35] H. Vargas, J. Sanchez Moreno, C. Jara, F. A. Candelas, F. Torres, and S. Dormido, "A network of automatic control web-based laboratories," *IEEE Trans. Learn. Technol.*, vol. 4, no. 3, pp. 197–208, Jul. 2011.
[36] E. Lindsay and M. Good, "Effects of laboratory access modes upon learning outcomes," *IEEE Trans. Edu.*, vol. 48, no. 4, pp. 619–631, Nov. 2005.

Martin Kalúz received the MSc and PhD degrees in the field of automation from the Slovak University of Technology (STU) in Bratislava, Slovakia, in 2010 and 2014, respectively. Currently, he is the postdoctoral researcher and lecturer at the STU. His areas of research are virtual and remote laboratories, embedded systems, software engineering and informatization of education process. He is a member of the IEEE.

Javier García-Zubía received the Graduation degree and the PhD degree in computer engineering from the Faculty of Engineering at the University of Deusto, Bilbao, Spain, in 1987 and 1996, respectively. He is currently a professor in the Faculty of Engineering at the University of Deusto. He is responsible for the remote lab at the University of Deusto (http://weblab.deusto.es). He is a senior member of the IEEE.

Miroslav Fikar received the ME and PhD degrees in chemical engineering from the Slovak University of Technology in Bratislava, Slovakia, in 1989 and 1994, respectively. He has stayed with the Faculty of Chemical and Food Technology at STU where he is currently a professor and institute director. He was postdoc fellow in Nancy, France (Elf Aquitaine, Ministére de la Recherche), Alexander von Humboldt Fellow in Bochum, Germany, and has spent several stays in Denmark, Germany, France, and Switzerland. His current research interests include optimal control, MPC, and chemical process control. He is a member of the IEEE.

Ľuboš Čirka received the MSc and PhD degrees in automation from the Slovak University of Technology in Bratislava, Slovakia, in 1996 and 2003, respectively. He is currently an assistant professor in the Faculty of Chemical and Food Technology at STU in Bratislava. His areas of research are virtual and remote laboratories, adaptive and robust control, and system identification.