Proceedings of the 2nd International Workshop on e-learning and Virtual and Remote Laboratories

Bernhard Rabe, Andreas Rasche (eds.)

Technische Berichte Nr. 21

des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam

Technische Berichte des Hasso-Plattner-Instituts für Softwaresystemtechnik
an der Universität Potsdam

Nr. 21

Proceedings of the 2nd International Workshop on e-learning and Virtual and Remote Laboratories

Bernhard Rabe, Andreas Rasche (eds.)

Potsdam 2008

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar

Die Reihe *Technische Berichte des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam* erscheint aperiodisch.

Herausgeber: Professoren des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam

Redaktion: Bernhard Rabe, Andreas Rasche

E-Mail: {bernhard.rabe; andreas.rasche}@.hpi.uni-potsdam.de

Verlag: Universitätsverlag Potsdam
Am Neuen Palais 10, 14469 Potsdam
Fon +49 (0) 331 977 4517
Fax +49 (0) 331 977 4625
e-mail: ubpub@uni-potsdamde
http://info.ub.uni-potsdam.de/verlag.htm

Druck: allprintmedia gmbH
Blomberger Weg 6a
13437 Berlin
email: info@allprint-media.de

Heft Nr. 21 (2008)
ISSN 1613-5652
ISBN 978-3-940793-17-1

Foreword

It is our pleasure to welcome all participants to VIRTUAL-LAB'2008, the 2nd International Workshop on e-learning and Virtual & Remote Laboratories, at Hasso-Plattner-Institute for Software Systems Engineering at University of Potsdam, Germany. The workshop is a successor of the successful VIRTUAL-LAB'04 workshop held in Setúbal, Portugal.
The aim of this workshop is to present and discuss the latest development in the area of remote and virtual laboratories. Among other topics the interconnections of heterogeneous laboratory infrastructures by using middleware will be one special topic of the workshop. The workshop is sponsored by the Leonardo Da Vinci Programme of the European Union within the VET-TREND-project (RO/06/B/F/NT175014).
These proceedings include 14 high-quality papers we received from 8 countries. Each paper has been reviewed by at least 3 members of the international program committee. As a result we are proud to present a collection of papers covering challenging topics, ranging from architecture of virtual & remote laboratories to service-orientation in virtual and remote laboratories, also including "New e-learning techniques" and "Analysis of virtual and remote laboratories in the field".
We would like to thank all authors who submitted papers and the Program Committee members for their efforts in properly reviewing the submissions. We also address our special thanks to Sabine Wagner who spent major effort around the workshop organization.
The success of any workshop depends mainly on the quality of the program and participation of people. We thank you all for being here. We wish that your stay in Potsdam is really enjoyable.

Bernhard Rabe **Andreas Rasche**

VIRTUAL-LAB'2008 Organizing Committee
Hasso-Plattner-Institute, University of Potsdam
Potsdam, Germany

Workshop Committees

Advisory Committee:
Andreas Polze *(Hasso-Plattner-Institute, Germany)*
Christoph Meinel *(Hasso-Plattner-Institute, Germany)*

Organizing Committee:
Bernhard Rabe *(Hasso-Plattner-Institute, Germany)*
Andreas Rasche *(Hasso-Plattner-Institute, Germany)*

Program Committee:
Ioannis Anagnostopoulos *(University of the Aegean, Greece)*
Paul Borza *(University TRANSILVANIA of Brasov, Romania)*
Fernando Coito *(Univ. Nova de Lisboa / UNINOVA, Portugal)*
Dirk Cordel *(Hasso-Plattner-Institute, Germany)*
Luis Gomes *(Univ. Nova de Lisboa / UNINOVA, Portugal)*
Leandro Soares Indrusiak *(Technical University Darmstadt, Germany)*
Eleftherios Kayafas *(NTUA, Greece)*
José Martins *(CINEL, Portugal)*
Bartolomeo Montrucchio *(Polytecnico di Torino, Italy)*
Sorin Moraru *(University TRANSILVANIA of Brasov, Romania)*
Domenico Ponta *(University of Genoa, Italy)*
Florin Sandu *(Siemens PSE, Romania)*
Anna Marina Scapolla *(University of Genoa, Italy)*
Gheorghe Scutaru *(University TRANSILVANIA of Brasov, Romania)*
Alexander Schmidt *(Hasso-Plattner-Institute, Germany)*
Michael Schöbel *(Hasso-Plattner-Institute, Germany)*
Wolfgang Schult *(Hasso-Plattner-Institute, Germany)*
Peter Tröger *(Blekinge Techniska Högskolan, Sweden)*

Contents

An Internet-Based Laboratory Course in Chemical Reaction Engineering and Unit Operations

Ralf Moros
Institut für Technische Chemie
Universität Leipzig
Linnéstr. 3
D-04103 Leipzig, Germany
moros@uni-leipzig.de

Frank Luft
Büro Dr. Frank Luft
Holsteinstr. 52
D-04317 Leipzig
luft@superkabel.de

Roger Gläser
Institut für Technische Chemie
Universität Leipzig
Linnéstr. 3
D-04103 Leipzig, Germany
roger.glaeser@uni-leipzig.de

Abstract

Practical courses are most important parts of studies in Chemistry. A laboratory course in Chemical Reaction Engineering and Unit Operations typically consists of several hands-on experiments. We present an online laboratory course for web-based distance learning. This online course consists of a web-based implementation of manuals for each experiment including animations and simulations as well as of remotely controlled experiments. An essential objective of the project was the set-up of interfaces for observing and controlling laboratory experiments via the Internet using a standard web-browser only.

A Java-based client/server method for such a remote control of laboratory experiments will be described. In addition, the structure of the web-based material will be explained and the first experience with this kind of web-based material and with distributed remotely controlled experiments will be reported.

1 Introduction

One of the most important elements of studies in natural sciences are practical laboratory courses. During the last years, numerous efforts were made to integrate real experiments into online material.

There are two possibilities to include experiments into web-based material. The first is the web-based simulation of an experiment. The second is to implement remote access into real laboratory equipment and real experiments [1]. A simulation is always based on a mathematical model and does not exactly reflect the "real" world. Therefore, remote access to real experiments is an interesting way to include experiments into online material.

There are several interesting applications for such remotely controlled experiments. One example is the heat transfer experiment [2] at the iLab project [3] of the Massachusetts Institute of Technology (Boston, USA). Another example is the non ideal reactor experiment [4] which is located at the University of Cambridge, Department of Chemical Engineering (Cambridge, UK). Some further applications of remotely controlled experiments in chemical engineering were discussed during the web lab workshop in Cambridge, UK, in 2005 [5]. During the last years many new remote labs have been established, especially in engineering, robotics and physics [6,7].

The e-Learning group of our institute [8] has developed remotely controlled experiments since 2001 [9] as part of the completed German project „Vernetztes Studium Chemie". We have developed web-based distance learning material for a practical laboratory course. This practical course consists of several learning units for experiments in chemical reaction engineering and unit operations. The primary user group are bachelor students of chemistry.

The users have access to more than one hundred learning units (LU) through the CHEMGAPEDIA portal [10], which is hosted by FIZ Chemie GmbH. These LUs can be used for lectures, seminars and practical courses or for additional individual education.

Using the practical course LUs, the students obtain information about an experiment before performing it. In addition, the students can carry out different experiments as interactive simulations and as on- and offline real experiments. Our practical course LUs offer explanations on theoretical concepts and can be used to assist in the evaluation of measured experimental data.

2 A practical course LU

The practical course LU was designed similarly to the real laboratory course.

This leads to a structure of the LU as shown in Figure 1.

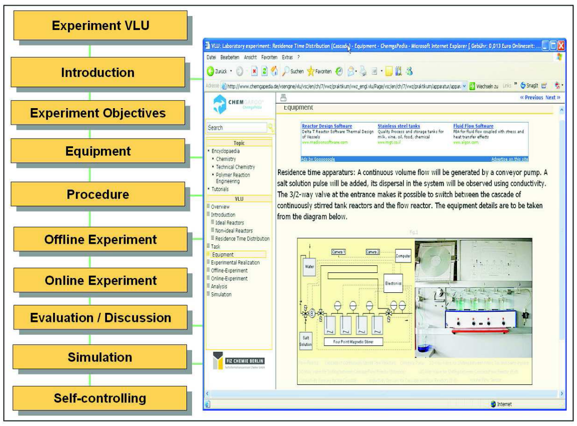

Figure 1: Structure of a practical course LU.

For each experiment an *Introduction* is available which describes the theoretical background of the experiment. This section is followed by *Experiment objectives* and a description of the hardware used for the experiment (*Equipment*). Interactive pictures are integrated to help understanding the hardware of an experiment. In the *Procedure* section, the students learn how to use the control software for controlling the experiment online.

Students apply different learning strategies. Some students might prefer interactive simulations of experiments, others prefer offline experiments or require carrying out the real experiment. Our LUs offer all these different approaches for conducting an experiment.

In the section *Online experiment* a student who completed all previous sections is enabled to carry out the real experiment.

The section *Offline experiment* offers the possibility to carry out an offline experiment. Based on recorded experimental data (data, video, audio), we created a web-based multimedia animation. This animation is embedded into the *Offline experiment* section and is delivered by a streaming server.

An interactive simulation (*Simulation*) provides the third possibility to carry out an experiment.

Valuable help for evaluation of the obtained data can be found in the section *Evaluation/Discussion.*

A section *Self-controlling* completes the LU. The students can carry out different tests to evaluate their knowledge.

3 The design of a remote experiment

From a student's perspective, all that is required to perform a remotely controlled experiment should be a standard Web browser that supports Java. Thus, the students are able to observe and control a remote experiment from different locations at any time.

Access to remote experiments is based on a client/server architecture for data acquisition and control written in Java as shown in Figure 2.

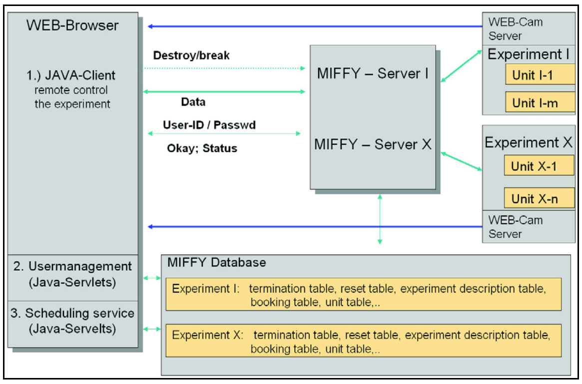

Figure 2: Client/server architecture.

The name of our data acquisition and control server is MIFFY-server.

User data, e.g., user ID and password, reservations and data of experiments, e.g., broadcast interval, IP addresses, number of hardware units, are stored within the MIFFY-database.

One MIFFY-server can control one online experiment. Each experiment consists of one or more hardware units such as units for digital I/O, analog I/O or serial I/O. For the data acquisition a TCP/IP-based hardware, called EDAS (Intelligent Instrumentation Inc. [11]), is used.
It is possible to follow the "live" experiment using real-time video and audio in addition to the data acquisition and controlling. However, the data acquisition and controlling are more important than the live stream, because the students have to evaluate the data, not the video.

From a developer's perspective, applet clients for new remote experiments are easily assembled using our library of Java Bean components (VJBL: VIPRATECH Java Bean Library): the developer creates the observation and control GUI (Graphical User Interface) using drag and drop functions. There are several virtual devices in the VJBL like digital meter, analog meter, chart recorder, buttons, digital I/O, analog I/O, etc. All libraries and the Java server will be published as open source.

The MIFFY-server is easy to configure by using a Java application.

4 Performing an online experiment

Before carrying out an experiment, the students receive a user ID and password for the chosen experiment. Based on this identification, the user can reserve a time for the planned experiment. Data for this scheme are stored into the MIFFY-database. A simple user management and scheduling software is implemented by using Java servlets.

The person who plans an experiment has to log into the MIFFY-server by using a web page. The user ID and password will be checked (Figure 3) after logging into the server.

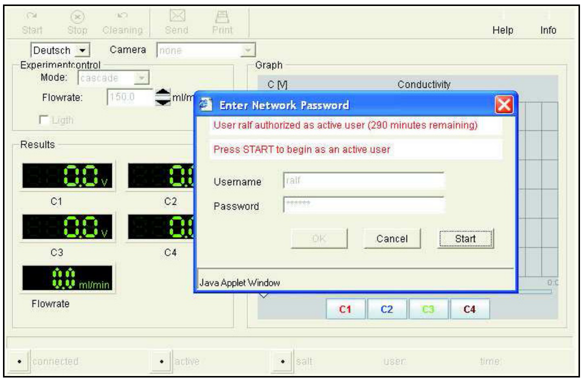

Figure 3: User and login interface.

If everything is correct and a time reservation for the experiment has been made, the students are given access to the experiment by the system. A Java applet for controlling and data acquisition is loaded into a web page. Figure 4 shows a typical web page for the remote control of an experiment.

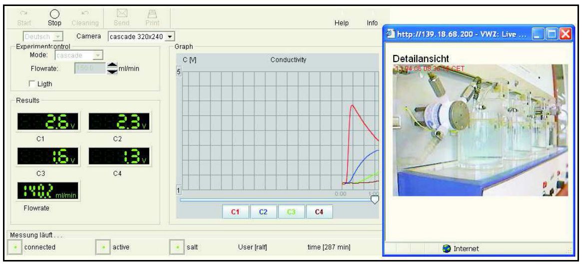

Figure 4: Data acquisition and real-time observation.

The users can change parameters of the experiment, start and stop the experiment, print out graphs, send the acquisition data via e-mail and activate real-time observation of the experiment. For the real-time observation, the students can choose the resolution of the web-cam. The resolution depends on the web-cam which is installed - generally, three different resolutions (192x144, 240x180, 320x240) are implemented.

Typically, the experiment is finished by closing the web page. In this case, the MIFFY-server terminates the experiment.

While one student controls an experiment, up to 20 other students can observe this same experiment.

5 First Experience

Based on these concepts of remote control, seven experiments located at two different universities (Leipzig and Oldenburg) have been developed since 2001:

- Heat transfer;
- Residence time distribution;
- Temperature control;
- Hydrolysis;
- Dehydration;
- Adsorption;
- Remote control demonstration (Energy house).

At the University of Leipzig, a new experiment (Residence time distribution in a real reactor system) will be completed within the next few months.
Four chemical institutes from different universities worked together to create another new experiment (Micro reactor system). The latter will be built up at the University of Oldenburg within the next six months.

Our experiments are included into the usual laboratory course (Laboratory Course in Chemical Reaction Engineering and Unit Operations) as well as into lectures at both Universities of Leipzig and Oldenburg. Experiments are also included into a practical course for further education. Access to the experiments was tested from different locations in Germany, Great Britain, USA, Korea, South Africa and New Zealand and under different conditions (high speed internet access, internet café's, WLAN). From every location and under all selected conditions, access to the experiments was shown to be successful.

The user has numerous advantages including:

- access to experiments from any location at any time;
- access to experiments from within an experiment learning unit;
- possibility to compare the real experiment with the simulations;
- possibility to carry out and repeat an experiment without being observed by a supervisor or assistant;
- possibility of self-control.

From the institutions' point of view there are additional advantages:

- developing and sharing of high-value lab resources between several institutions;
- bringing lab experiences into the lecture hall;
- allowing more flexible timetabling of labs;
- using the lab resources more efficiently.

The student's reception of the learning units and of the remote experiments has been evaluated by using questionnaires. The evaluation results are generally positive. The students are satisfied both with this kind of experiments and with the learning units. They appreciate the possibility to do the experiment from a place and at a time they have chosen by themselves. They are using the learning units for information about the theoretical background, the experimental setup and the data evaluation before actually carrying out the experiment. The students do not experience problems to manage the user interface of the remote experiments.

6 Conclusions

Experiments or laboratory devices with the possibility of remote access and control are a very powerful tool in chemical engineering education. They enable both students and the supervisors to perform and use experiments independent of time and location. It is possible to share experiments and laboratory devices between institutions based on a network of remote control experiments. The sharing of resources and the exchange of know-how as well as the reduction of costs for the experiments are an attractive result of such a network.

References

[1] J. Bourne, D. Harris, F. Mayadas: Online engineering education: Learning anywhere, anytime. *Journal of Engineering Education*, 94(1), 2005

[2] MIT iLab Heat Exchanger; http://heatex.mit.edu/

[3] MIT iLab project; http://icampus.mit.edu/projects/iLabs.shtml

[4] University of Cambridge, Department of Chemical Engineering, Weblabs; http://como.cheng.cam.ac.uk/weblabs/index.html

[5] A. Selmer, M. Kraft, R. Moros, C.K. Clark: Weblabs in Chemical Engineering Education. *Education for Chemical Engineers*, 2(1): 38-45, 2007

[6] S. Gröber, M. Vetter, B. Eckert, H.-J. Jodl: Experimenting from a distance – remotely controlled laboratory (RCL). *European Journal of Physics*, 28(3): 127-141, 2007

[7] S. Jeschke, Th. Richter, H. Scheel, Ch. Thomas: On Remote and Virtual Experiments in eLearning. *Journal of Software*, 2(6): 76-84, 2007

[8] Institut für Technische Chemie, Universität Leipzig; http://techni.chemie.uni-leipzig.de/

[9] R. Moros, F. Luft, H. Papp: Virtual Laboratory Course in Chemical Engineering and Unit Operations (VIPRATECH) Tutorials, Simulations and Remote Process Control. *2002 Proceedings of the International Conference on Computers in Education*, Auckland, New Zealand

[10] CHEMGAPEDIA: www.chemgapedia.de

[11] Intelligent Instrumentation Inc., EDAS: http://www.edasce.com/ethernet-data-acqusition.html

Internet Based Laboratory for Experimentation with Multilevel Medium-Power Converters

Carlos Girón
Electronic Engineering Department.
Alcalá University, Madrid, Spain
carlos.giron@depeca.uah.es

Francisco Huerta
Electronic Engineering Department.
Alcalá University, Madrid, Spain
francisco.huerta@depeca.uah.es

Fco. Javier Rodriguez
Electronic Engineering Department.
Alcalá University, Madrid, Spain
fjrs@depeca.uah.es

Emilio Bueno
Electronic Engineering Department.
Alcalá University, Madrid, Spain
emilio@depeca.uah.es

Julio Pastor
Electronic Engineering Department.
Alcalá University, Madrid, Spain
pastor@depeca.uah.es

Abstract

This paper describes the hardware and software modules of a remote Internet-based laboratory for experimentation with a 120KVA multilevel power converter. The key feature of this laboratory is the utilization of a multilevel converter that can be controlled and supervised remotely in a secured way. A wide variety of laboratory experiments are supported, from grid connection to ac motor control. The users can chose the control structure, the control parameters and the kind of load and get the graphical results of the measurements, all in real time.

1 Introduction

The remote access to complex laboratory equipment and the possibility of remotely driving experiments and measurements represent appealing issues of great interest and relevance for research, educational, and industrial purposes. The range of topics potentially involved is very large, including, among others, applications in all fields of engineering, as well as of medicine, biology, and physics [1]–[4].

Significant experiences in this field are now propelled by the availability of a large set of programmable instruments and by the rapid growth of networking infrastructures.

Several works about remote laboratories have been published. Poindexter and Heck [5] explain in a tutorial, among other things, how to create WWW sites dealing with internet learning and teaching, presenting some models, examples and specific applications about remote-laboratories, like an automatic analogical-circuit test-system [6], a remote measurement-laboratory [7], a robotic control [8] and a chemical process controlling equipment [9].

The aim of this paper is to explain the design and implementation of a remote laboratory intended for experimentation with medium power multilevel converters. These converters present two fundamental advantages compared to others topologies: (1) with the same power semiconductors the converter nominal power will be higher and (2) the harmonics after the fundamental are placed at higher frequencies.

With the practices developed in this laboratory, the users can understand the main functional principles of these converters. Also, they can configure and supervise some of the typical applications of this kind of converter as: AC motors excitation, management of energy generated by wind turbines, active filters, grid synchronization, power quality, etc.

Compared with other similar remote laboratories that are indicated for under graduated students, this one, due to its capabilities, is intended for post-graduated students and for power electronic engineers.

After this introduction, the paper is continued with section II, which explains the architecture of the hardware and the communications elements that the laboratory implies. In section III, the user interface and its supporting hardware and software modules, are explained. Section IV contains, as example, the development of one practice that can be completed with this laboratory. Finally, the conclusion can be found in section V.

2 Hardware Architecture

2.1 Block Diagram

The general system block diagram is shown in Fig.1. Its final aim is to make possible the realization of remote laboratory practices with a multilevel power converter. One of the main requirements of the system is that the final user needs no specific software or hardware. Therefore, a web based interface is used together with the free plug-in “LabView Real-Time Engine” [10]. The environment LabView is provided with a web-publication system of the virtual instruments (VIs), which gives the user the chance to control remotely the running VI.

The host is used as interface between the client, connected via internet, and the final system. An interface USB 2.0 [11] is used between the host and the control platform. It provides the necessary high speed to transfer all the sampled signals. The host application, programmed in LabView [10], is the responsible of the control panel publication by means of a web server embedded in the development environment. With this application is possible to monitor the system, select the control scheme and send references and control parameters to the control platform.

The control platform is composed by several subsystems:

A floating-point DSP TMS320DSK6713 of Texas Instruments and a FPGA Spartan 3 of Xilinx make up the Processor Subsystem. Their aim is to implement the programmed control tasks.

The acquisition subsystem is composed for 6 ADCs of 4 channels and 12 bits per channel to acquire the analog signals of the converter; 9 fiber-optic receivers to implement the reading of 3 encoders; and 12 fiber-optic receivers to monitor possible errors in the IGBTs operation.

The actuation subsystem is composed of 24 fiber-optic transmitters to send the PWM signals; and 7 general purpose I/O to connect/disconnect the contactors of the converter.

The control platform [12] is shown in Fig. 2. This board provides the hardware support to implement the control, and supervision of the system, including the signal acquisition and conditioning. Also, this platform offers USB connectivity. The PWM signals are sent from the control platform to the IGBT drivers through optic fiber. This control platform is also responsible for the actuation over the commutation matrix, which connects as load an ac motor or a passive load, according to the user selection.

As shown in Fig. 3, the Power Electronic chosen topology is a back-to-back NPC (three levels) VSC. The power consumed or generated by the induction motor is rectified to DC by the converter VSC2 that also supplies lagging excitation current to the machine. The DC-bus power is inverted by the VSC1 and driven to a utility grid at a programmable power factor. Fig. 4 shows the block diagram of the control system, which is divided into two main controllers: the line-side converter and the generator-side converter. In Fig. 4, the line-side converter (VSC1) is vector controlled, using direct vector control and synchronous current control in the inner loops. The grid power is controlled with DC-bus voltage control. On the other hand, the generator-side converter (VSC2) controls the generator speed by indirect vector control with torque control and synchronous current control in the inner loops. The machine flux is controlled in an open loop by the control of the current Ids but, in normal conditions, the rotor flux is set to a rated value for a fast transient response. Any case, the system permits other control configurations that will be analyzed in the section III.

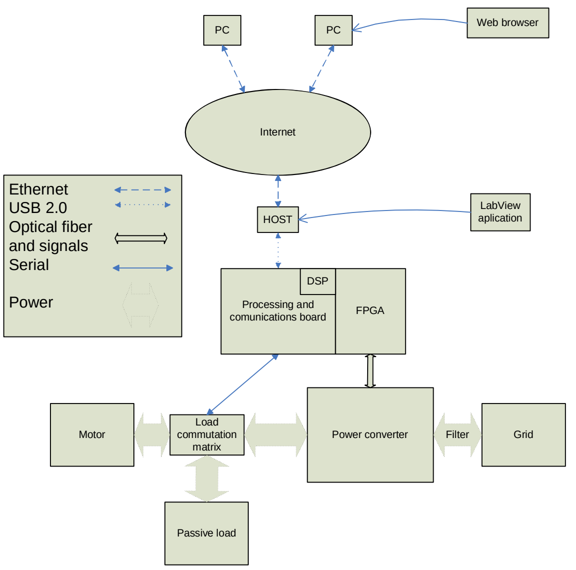

Fig. 1, Block Diagram

Fig. 2. Processor Board

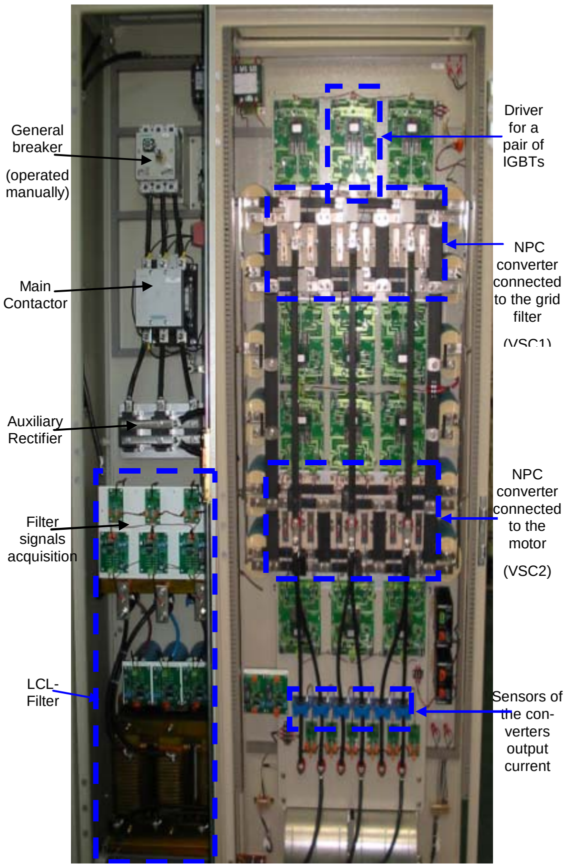

Fig. 3. Power Electronic System of the converter.

2.2 Protection capacity

The system is provided with protections in view of a bad operation (e.g. signals saturation) and an erroneous user configuration (e.g. out-of-range references). These protections are distributed between the control platform and the host:

The FPGA informs about bad operations of the IGBTs and out-of-range measurements in the ADCs.

The DSP avoids controller variables' overflows. Also it detects when the variables operation ranges are exceeded.

The host avoids sending references that could destroy the system components exceeding the operation limits. The error logs are shown both in the host and in the user terminal.

3 Interface Description.

3.1 Host Interface

The application interface that runs on the host, has been developed using LabView. This application can be accessed by a remote terminal through the web service supplied by LabView. So, it is possible to control the system using a web navigator, with a client-server communication architecture

Into the application there exist two panels: configuration panel and representation panel.

In the configuration panel, shown in Fig. 5, the operation mode of the system can be set. VSC1 can be configured as active filter or as active rectifier, connected to a passive load or to a active load (VSC2 works). Moreover, the filter to use (L or LCL), should be selected.

Once this first configuration is finished the converter block diagram is showed in the panel, attending the selected options. Next options to configure are the kind of VSC control. VSC1 control can be

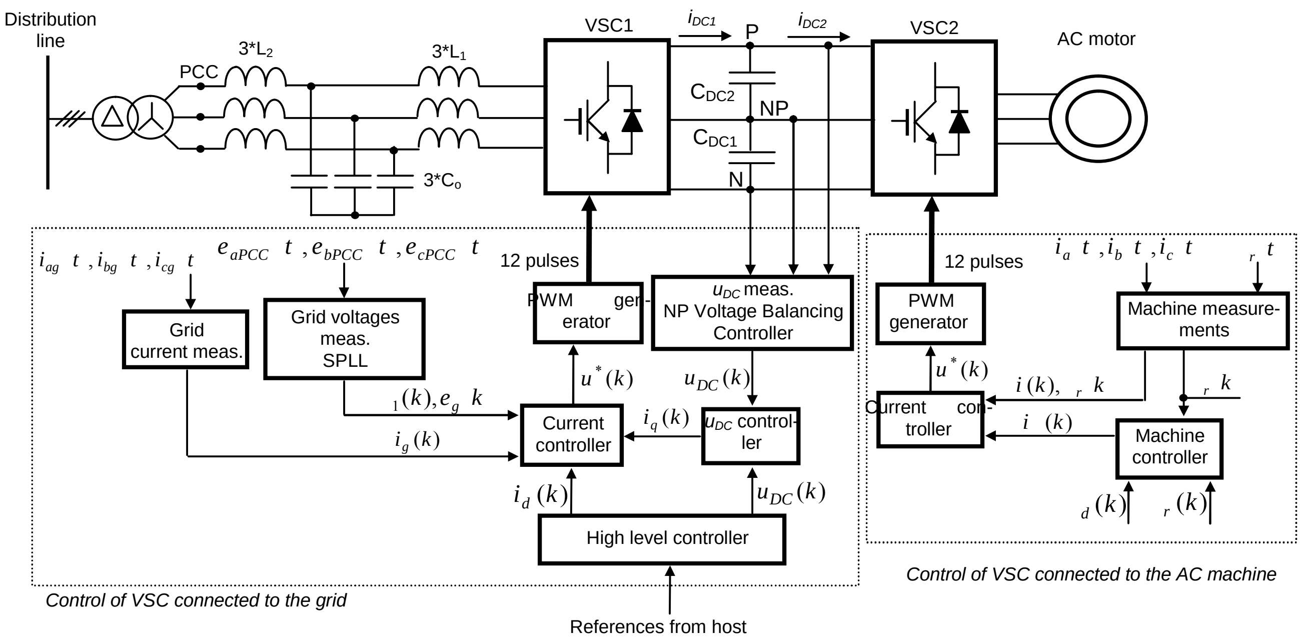

Fig. 4. Control block diagram of the wind generation system.

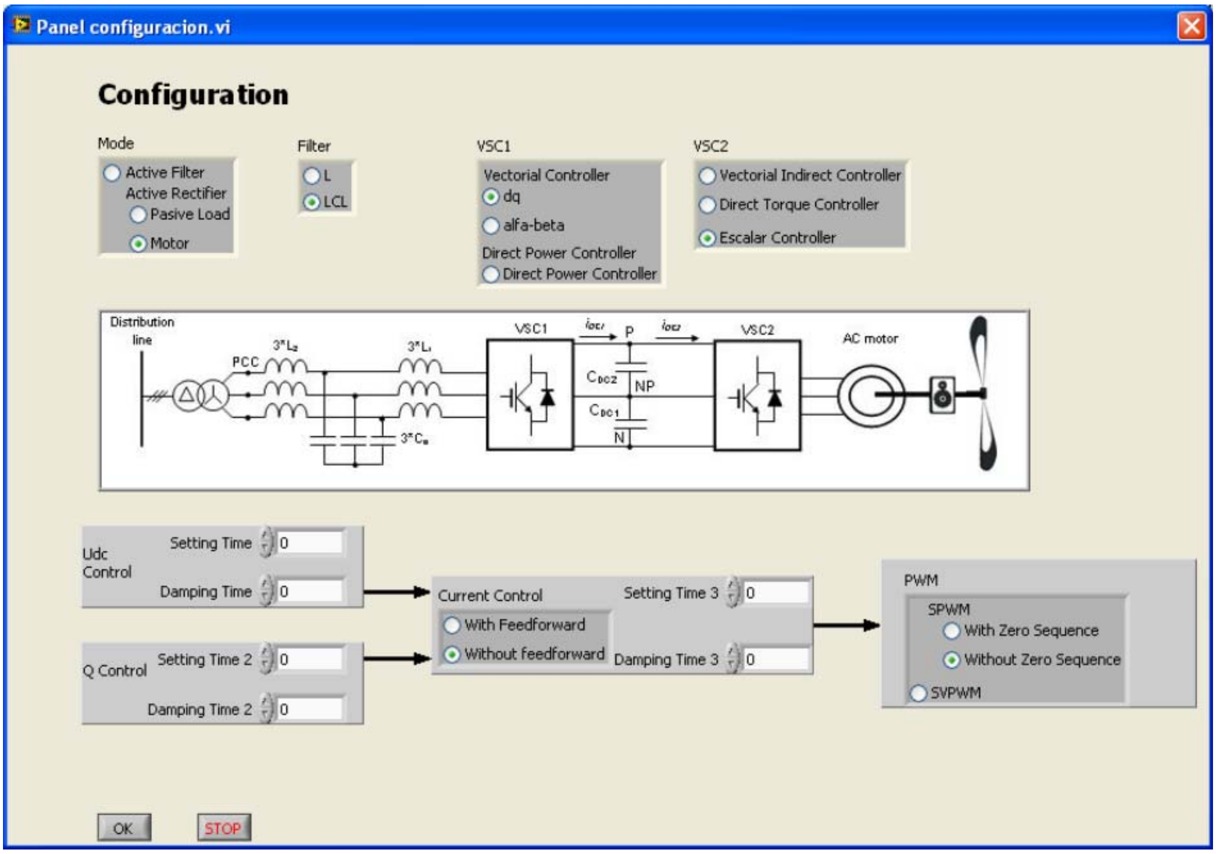

Fig. 5, Configuration Panel

selected as dq or αβ frames vectorial controller or direct power controller (DPC). VSC2 control can be chosen as vectorial indirect controller, direct torque controller (DTC) or scalar controller [13]. When the kind of control has been chosen, a block diagram is presented with the control parameters that can be configured by the users.

When the converter configuration is ended, the experimental plots screen is showed. In this panel, see Fig. 6, the converter operation is monitored. When the user has introduced the references, the start button should be pushed. In this moment, the system opera-

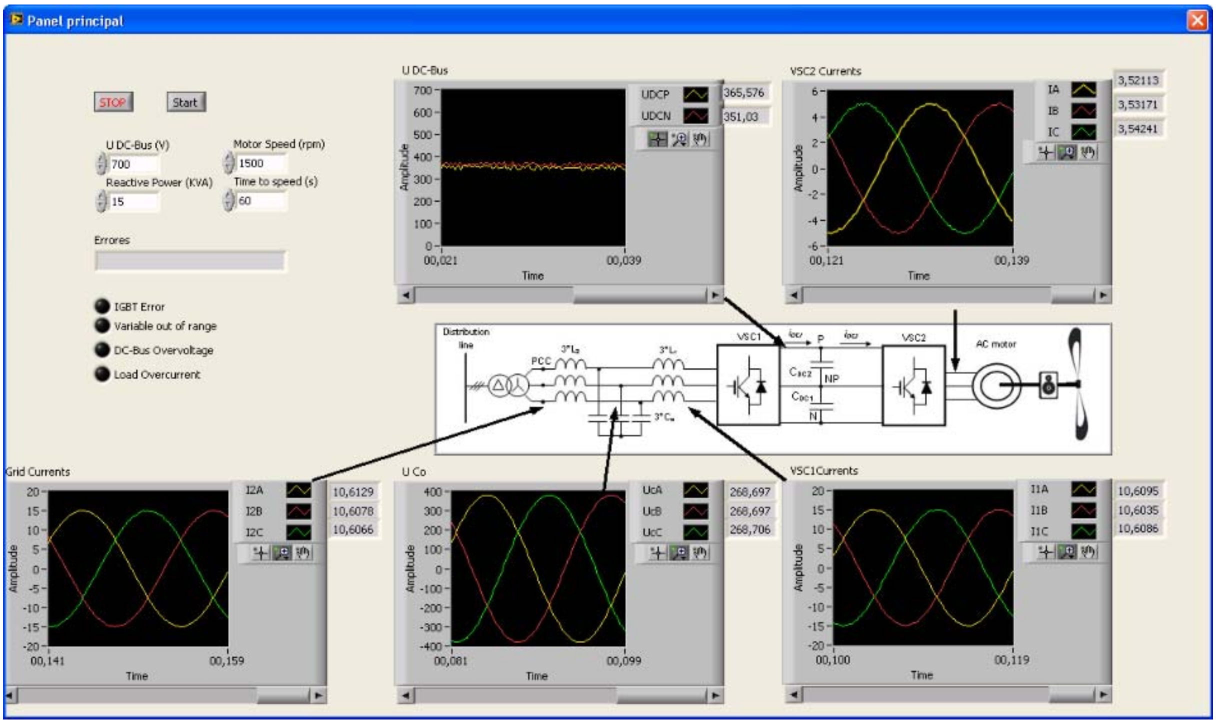

Fig. 6, Representation Panel

tion starts, showing in the graphics the most meaningful signals, e.g. the dc bus voltage, the motor currents or the grid currents. These graphics can be resized on-line. The application is able to save all the samples in a text file, which can be downloaded via web. Also, the signals to save can be chosen.

If an error occurs, the application informs to the user with an explanatory error message and light on the appropriated indicator. In this way, a configuration error or a system failure is easily detectable. Clicking over the error indicator, coming back to the configuration panel, where rearmed system parameters are marked. From here, it is possible to stabilize or reinitiate the system. When the application is closed, the system makes a secure stop, independently of the way that the user stopped it. When it is stopped, it is rearmed and checked to be available for a new user.

3.2 Host-Processing board communication description

The host-board communication is carried out by means a USB connection, with a transfer speed of 480 Mbps, using a bandwidth of 20MB/s.[12]

The processor and acquisition card, contains a FPGA, DSP, a set of ADC´s and transceivers of optical fibber [14].

USB communications between the processing card and host, provide the capacity to maintain a shared memory between the DSP and the host so, it can monitor the control system variables and parameters, and therefore, it can send this information across local network or Internet to any other host or remote terminal. These communications are handled by a Cypress CY7C68001 SX2 controller [15], which provides a parallel interface for the DSP connection. DSP uses EDMA module to handle transferences between the internal memory and USB controller, thus releasing the DSP CPU of these operations, so that the control algorithms execution is not delayed.

This communication message-based operates by commands defined by the application. First, the converter and control configuration are sent. At the end, a start command is sent to the control platform to start the execution of the configured control loops.

Signal samples are sent as frames from the control platform to the host at the end of the control loop through the EDMA of the DSP. In this way, it is possible to send the data without using the DSP's CPU.

When the frames are received, the data are extracted and plotted in screen. The parallelism used in the application make possible to save data in files while they are being represented. The data are saved in Matlab format for their off-line analysis and make easier the documentation process.

4 Practice Cases

In this section, one practice case developed with the described laboratory is presented. VSC1 is connected to the grid through LCL filter, and the VSC2 is driven to an induction machine. The indirect vectorial controller is used to control the second converter. To carry out the experimental tests described here, the converter has been connected to a real grid with nominal voltage of 400V in the laboratory; and, furthermore, with the next conditions: (1) non-null grid impedance; (2) mean THD in the phase neutral voltages of 19.5 %; and, (3) mean percent imbalance of 3.5% (according to IEEE Std 1159-1995). Fig. 7 shows the grid currents and voltages when the VSC2 is approximately consuming 12kW and the reactive power reference of the VSC1 is $Q^* \quad 0VAr$ (see Fig. 7.a), $Q^* \quad 8kVAr$ (see Fig. 7.c), and $Q^* \quad 8kVAr$ (see Fig. 7.d). In Fig. 7 the angle between the voltage and the current is 180º because the converter is consuming active power from the grid utility.

In the contrary case, if the converter is delivering active power to the grid utility, the current and the voltage will be in phase. Fig. 7.b shows the grid current harmonics for the test shown in Fig. 7.a. The fundamental harmonic peak value is $\frac{12kW}{400}\sqrt{\frac{2}{3}}$, and the value of the other harmonics are negligible, including the harmonics placed at $2 f_{SW}$. This verifies the design procedure of the grid filter and, also, the VSC1 current controller algorithms. For the two other experiments, the harmonics are very similar: only the fundamental harmonic amplitude increases 5A approximately.

Fig. 8 shows the temporal evolution under different DC-bus voltage references (u_{DC}), and different reference levels of active and reactive powers. This signal has been represented with Matlab from the data storage in the text file during the operation of the converter.

Initially, an auxiliary rectifier makes the precharge of the DC-bus capacitors until the natural DC-bus voltage ($\sqrt{2}U_{base}$); and from this point, the VSC1 controller begins to work. The first step is that the DC-bus reaches the u_{DC} through a process called 'soft-start', which consists of applying a slope reference of value $u_{DC} \quad \sqrt{2}U_{base} \, {}^{V}\!/_{s}$.

Then the DC-bus controller with its feed-forward action begins to normally operate. In spite of the different references of active and reactive power applied, implying jumps in the DC-bus voltage response, u_{DC} does track to u_{DC} without observable errors.

This tracking process verifies the correct operation of the DC-bus controller with its feed-forward action.

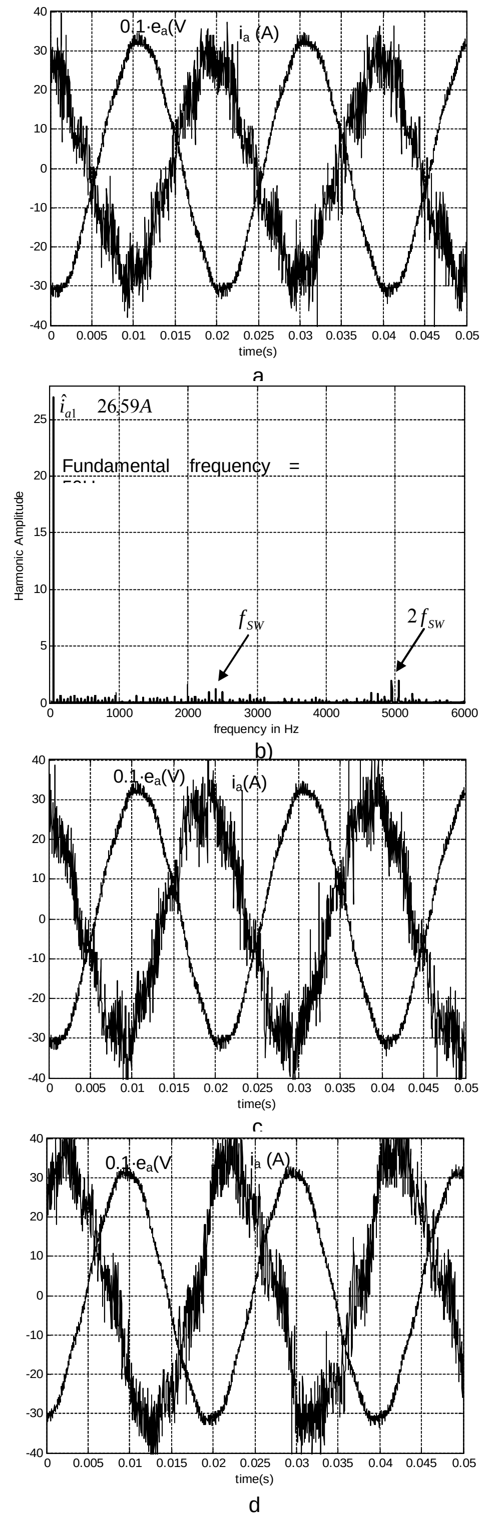

Fig. 7. Experimental converter waveforms. a) Phase voltage divided by 10 and phase current for $Q^{*} \; 0VAr$. b) Phase current harmonic spectrum for the previous condition. c) Phase voltage divided by 10 and phase current for $Q^{*} \; 8kVAr$. d) Phase voltage divided by 10 and phase current for $Q^{*} \; 8kVAr$.

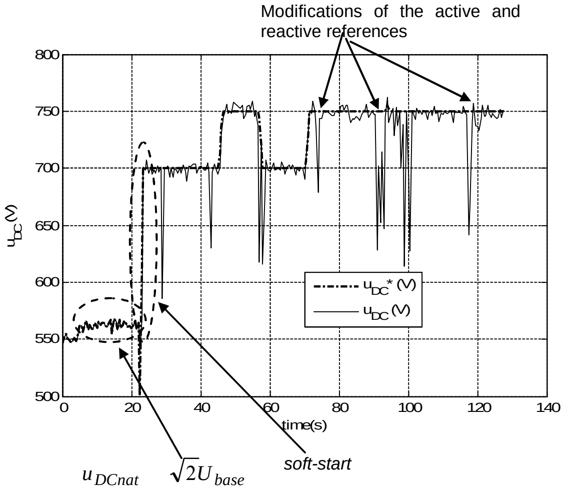

Fig. 8. Experimental waveforms of u_{DC}.

5 Conclusions

In this paper has been explained the implementation of a remote experimentation laboratory appointed to control power converters. The system autonomy is ensured by means of the fault protection subsystem, which provides reset and reactivation options. The implemented applications verify the next objectives: (1) to facilitate the laboratory experimental tests with a three-level NPC power converter in remote mode; (2) to promote the teaching of this kind of converters to post-graduated students; and (3) to facilitate to power electronics engineers the design and test of control algorithms.

Acknowledgment

This work has been funded by the Spanish Administration (MEC: ENE2005-08721-C04-01).

References

[1] M. J. Finley, “Tele-learning: The “killer app”?,” Guest Editorial, IEEE Commun. Mag., vol. 37, no. 3, pp. 80–81, Mar. 1999.

[2] S. Hackbarth, “Introduction to special issue on web-based learning”, Educ. Technol., vol. 37, no. 3, pp. 5–7, May–Jun. 1997.

[3] Y. L.Wu, T. Chan, B. S. Jong, and T.W. Lin, “A web-based virtual reality physics laboratory,” in Proc. 3rd IEEE ICALT, Athens, Greece, Jul. 2003, p. 455.

[4] D. Gillet, H. A. Latchman, C. Salzmann, and O. D. Crisalle, “Hands-on laboratory experiments in flexible and distance learning,” J. Eng. Educ., vol. 90, no. 2, pp. 187–191, Apr. 2001

[5] S.E.Poindexter and B.S.Heck, "Using the Web in Your Courses: What Can You Do? What Should You Do?," IEEE Control Systems Magazine, vol. 19, no 1, pp. 83-92. february 1999.

[6] D.Knight and S.P.DeWeerth, "World Wide Web-Based Automatic Testing of Analog Circuits," Proc. 1996 Midwest Symp. Circuits and Systems, pp. 295-298. August 1996.

[7] P.Arpaia. A.Baccigalupi. ECennamo and P.Daponte. "A remote measurement laboratory for educational experiments." Measurement. Vol 21 no 4, pp. 157-169, August 1997.

[8] A.Bhandari and M.Shor, "Access to an Instructional Control Laboratory Experiment Through the World Wide Web," Proc. ACC, Philadelphia. pp. 1319-1325,june 1998.

[9] M.Shaheen. K.Loparo and M.Buchner. "Remote Laboratory Experimentation", Proc. ACC. Philadelphia. pp 1326-1329, June 1998.

[10] NI LabView. National Instruments. (www.ni.com).

[11] USB Implementers Forum. (www.usb.org)

[12] C. Girón, F. J. Rodríguez, F. Huerta, E. Bueno, "Implementing high speed communication buses for a FPGA-DSP architecture for digital control of power electronics" in 2007 IEEE International Symposium on Intelligent Signal Processing (WISP'2007), Alcalá de Henares, Madrid (Spain), 3-5 Oct 2007, ISBN 1-4244-0830-X.

[13] M.P.Kazmierkowski, R.Krishnan, F.Blaabjerg. "Control in Power Electronics". Academic Press Series in Engineering. 2002.

[14] E. Bueno, S. Cobreces, F.J. Rodríguez, A. Hernández, F. Espinosa, R. Mateos, J.C. García, F. López. "Optimized design of a back-to-back NPC converter to be used as interface for renewable energies". 31th Annual Conference of the IEEE Industrial Electronics Society (IECON 2005), Raleigh, North Caroline (USA), 6-10 Nov. 2005. Page(s): 2543-2548. ISBN: 0-7803-9252-3.

[15] Cypress CY7C68001 EZ-USB SX2 datasheet (www.cypress.com)

Content management and architectural issues of a remote learning laboratory

Markus Wulff Patrick Lauer Torsten Braun

University of Bern, IAM
Neubrückstrasse 10
3012 Bern, Switzerland
{mwulff|lauer|braun}@iam.unibe.ch

Abstract

Internet-based distance learning requires an efficient development and management of learning material. It should be re-usable and it is necessary to keep the content in an open and portable format.

The infrastructure is expected to be available day and night. Users may access the experimentation setup from every Internet-connected work place at any time. Therefore, reliable hard- and software configurations must be used, which demand a low management overhead at the same time.

This article presents concepts and implementation that solve these issues for a remote hands-on networking laboratory. Concepts for reducing the management overhead, and increasing the availability of a remote laboratory through virtualisation are presented. Furthermore, a content formatting tool to support sustainability of the course material is introduced.

1. Introduction

Remote teaching and distance learning are of increasing importance for the education not only at universities. Online learning modules are an integral part of many courses and complement the traditional lectures. Such learning modules contain text, graphics, animations, and hands-on exercises via remote laboratory access.

The advantages of this blended learning approach are manifold. The students can access the online material from every place where they have an Internet connection and are not limited to fixed laboratory schedules. Furthermore, full online courses can be provided for students of remote learning universities or similar institutions. The students can read the relevant texts and do the laboratory exercises on-line, which provides a much higher flexibility. Even sharing courses with other institutions or providing access to other user communities is possible with an appropriate infrastructure.

In recent years, remote learning modules have been used to complement the lectures for the Bachelor and Master program at our institution. Most learning modules and infrastructure components have been developed during several e-learning projects [4, 17]. Due to a modular course architecture, adding new modules to the courses is fairly easy. The module repository can be completed step-by-step, and up-to-date research results can be incorporated.

In order to ensure sustainability of the course material, it must be updated from time to time and stored in an appropriate repository. Updates should be done only in this repository to prevent inconsistencies between the stored content and the content published in a Learning Management System (LMS). Furthermore, the text sources must be kept in a generic format, which is independent from the LMS used.

We do not host a LMS at our institute to keep the course content for the theory part—it has been outsourced. However, the laboratory infrastructure is installed and maintained in-house. The general course architecture is shown in Fig. 1. It has slightly changed during the years. The local access control system, which was used in earlier days, has been replaced by a Swiss-wide Authentication and Authorisation Infrastructure (AAI) [1, 13]. AAI is an attribute-based access control which enables a single sign-on system where the user is authenticated once at his or her home organisation. Then, checking authorisation of requested resource accesses is based on the user attributes the users home organisation provides. Thus, all users from any AAI enabled institution can register for an AAI enabled course by using their home organisation's account instead of getting a new user account locally.

With the reservation system, students can book time slots for each of the laboratory modules. If they want to access a module with a limited number of hardware devices, they get authorised by the reservation sys-

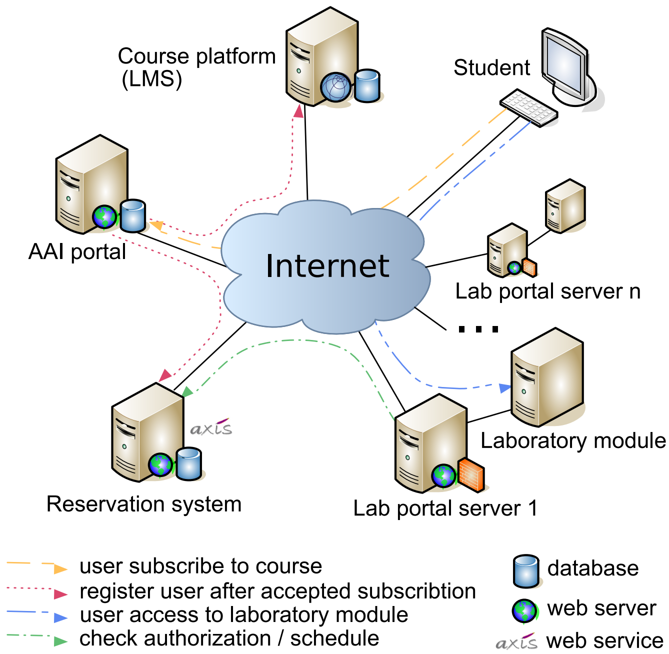

Figure 1. Current architecture of the e-learning laboratory infrastructure.

tem if the module was booked before by the respective user. The reservation for a certain module can be done over a Web based interface.

In the laboratory architecture, hands-on session modules are connected by a gateway that is called portal server. Portal servers connect the module specific hardware on one side and the user's computer on the other side. When the user wants to access the laboratory exercise, he has to authenticate himself at his AAI home organisation, which then sends attributes of the user to the portal of the respective resource. Based on this information the portal decides whether or not to grant access to the laboratory. Usually, there is one portal server per module but it is also possible to connect more than one module to a portal server. Any device that has an interface to a computer (for example a serial interface) can be connected to a portal server.

The hands-on experiments in our laboratory are not only used by local students. Other universities use the infrastructure as well for their education. This requires a high availability of the laboratory equipment. Not only the experimentation devices but also the portal servers and the reservation system must be up and running at any time. In contrast to these requirements, the administrative overhead running the exercise laboratory must be kept as low as possible.

In this paper, solutions for the issues of content management and laboratory architecture are discussed. In Section 2 a short overview about course content management is given. Furthermore, a tool for content formatting is introduced, which can convert learning module content into a specific format required by the LMS used. Section 3 then discusses architectural improvements of the laboratory infrastructure to solve the availability and management issues mentioned above. Finally, a short summary concludes the paper.

2. Online course content management

2.1. Overview and related activities

Several learning modules have been developed for our e-learning projects. One of the most important design goals of these modules is sustainability. In many cases the modules explain fundamentals of computer networks, operating systems or other computer science basics. Therefore, they are not subject to frequent updates and the content can be used for lectures over many years. Only smaller updates and corrections are performed if necessary.

The creation of such a learning module usually takes several weeks. Text must be written, graphics and animations must be created. It is obvious that this content should be stored in a Learning Management System (LMS) independent way to ensure that it can be used with any LMS. This especially applies if the LMS is hosted by a third party as it is not guaranteed that the current product is supported over the next years. In the case that a course must be migrated from one LMS version to another or, worse, to another product this will cause a lot of time consuming manual work.

The problem of storing e-learning in a reusable way has been already discovered several years ago. Different solutions have been proposed. The Sharable Content Object Reference Model (SCORM) [12] defines

a specific way of constructing Learning Management Systems and training content so that they work well with other SCORM conform systems. The common goals of different versions of SCORM are as follows: 1) packaging content and 2) exchanging data at run-time. The drawback of SCORM is that it is not supported by every LSM or the older versions of this standard only.
Many academic and other institutions are trying to make better use of networks and databases to efficiently and effectively achieve learning goals. One of the possible ways to go is to make learning resources accessible to educators and learners through learning object repositories (LOR). LORs are repositories, which organise reusable learning objects like courses, modules, but also images, videos and text documents in a clearly arranged way. To enable search engines to efficiently identify learning objects, a descriptive set of metadata is assigned to every object in the LOR. The goals here are re-usability of e-learning content, long-term archiving as well as to share teaching activities and make them visible to peers and to the public. Examples for LORs initiatives are the Switch LOR [9] in Switzerland or the eduSource project [8] in Canada.
If an appropriate LOR is provided in Switzerland we will consider using it. Not only on Swiss level such initiatives are on their way. A local LOR is also being developed at the University of Bern but this will depend on a specific LMS that we are not using today. This again shows the necessity of maintaining learning content in a platform independent way to be prepared for the frequent changes in today's e-learning landscape.

2.2. Content management

A satisfying solution for the exchange problem of learning objects between different content management platforms is not available until now. As stated in the Edutech LOR feasibility study [9], the interoperability is not guaranteed even if standards compliant content packages in SCORM format are exchanged. The integration of WebCT Vista LMS [3]—which is currently used for our courses—with any LOR causes problems because the API only allows to export bare content files without any course structure or other important metadata.
From today's perspective it is advisable to store our learning modules in a generic format and to set up a simple but efficient content management. The solutions to this content management issue should meet the following general conditions:

- The existing text sources can be integrated without significant modifications.
- New content can be created with tools that are available in our institute and people are familiar with (WYSIWYG HTML editors, word processors with HTML export capabilities etc.).
- It must be possible to create the content format required by a specific LMS on demand.
- The format of the learning content allows a later adaption to e-learning standards if required.

In order to be able to synchronise the access to the repository and to track modifications, the use of a version control system is advisable. Several solutions exist for this purpose. Subversion (SVN) [7] is one possibility here. It manages files and directories, and the changes made to them, over time. This allows to recover older versions of the repository data, or examine the history of how the data changed.

2.3. Content formatting

At an early stage of our learning content development it became clear that a tool for automatic content formatting is necessary. The goal was to reduce the effort needed to develop the learning content and to automatically create the pages for the LMS in an appropriate format. As a result the FFGF (file framework generator and formatter) [16] has been developed.
The FFGF tool is able to create the required document structure for a learning module including the header and footer for the pages. Furthermore, predefined templates can be provided for sections with general information, which are automatically included in the document. In a first step, the author has to create the table of contents for the module. This includes the definition of the time a learner should approximately spend for the sections. The time information is used to generate the module schedule to give the learner a clear overview over the time limits intended by the author. Figure 2 shows a sample schedule. In a second step, the FFGF tool creates the document structure by generating a separate file for each section and inserting the text from the templates provided. Now, the module author has to fill in the content into the section files (text, pictures, animations). This is done by using HTML as markup language. In the last step FFGF is run again and generates now the output, which can directly be uploaded to the LMS. The tool currently supports the format for WebCT CE and WebCT Vista.

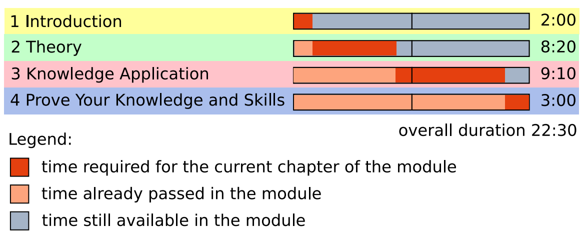

Figure 2. Course schedule example.

After several years of using FFGF we found some weak points in its architecture. The main problem is the fixed output format. It would require major changes in the program to adapt its output generator to another LMS document format. Another issue to be improved is the predefined document structure. This impedes subsequent changes to the document.

Due to the variety of mandatory changes to the current FFGF version, a complete redesign of the tool became necessary. The main improvements of the new programme version must be

- Easier adaptation to different LMS platforms
- Portable solution, usable by other institutions
- Better support for document management
- Simpler document syntax for easier document writing

Therefore, one of the main features of the new FFGF2 is its clear modular architecture. For the module author the most important feature is the automatic document structure detection and creation of the table of contents. It will no longer be necessary to define the module's table of contents in advance. The learning module content can be written into either a single file or split into several source files.

By keeping HTML for the markup of the text the writing of learning content can be done with a variety of available editors. The language is well known and provides all necessary features like different font styles, document structure and allows to include figures and animations. It is not the goal to provide yet another text/HTML editor nor an integrated development environment.

Figure 3 shows the architecture of FFGF2. The first layer represents the HTML document. For some sections, like generic introductions, templates are provided. The content of these sections is the same for every module and does not need to be rewritten every time. The HTML processing engine analyses the document structure and generates a XML [5] formatted document. At this point the templates are integrated into the document.

In the next layer the document is stored in XML format. This allows the content to be kept in a generic manner and simplifies the conversion to other formats. The XML format seems to be the most useful format regarding a later adaption to e-learning standards and the inclusion of learning object metadata.

The last component of FFGF2 is the output formatter. It has a generic interface combined with a plug-in mechanism. The respective plugins to create a certain output format can be attached here. Future changes in the output format do no longer require changes in the FFGF2 core. If a learning module must be formatted for a new LMS architecture or other document formats like PDF all necessary code is encapsulated in the output plugin.

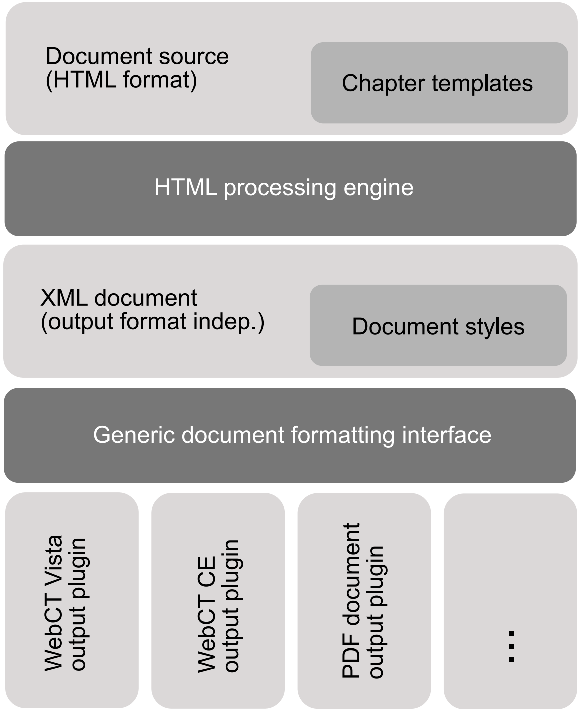

Figure 3. Architecture of the content formatting tool.

Once the document has been generated it is uploaded to a LMS like WebCT. A learner who has registered for the specific course is able to read it there, see the included pictures, watch the animations etc. The learning module document usually contains the description of the hands-on experiments as well. The laboratory exercises, however, are not stored on the LMS. They are located in our laboratory. A link in the document leads the reader to the laboratory computer for the respective experiment.

3. Laboratory infrastructure

As already described in the introduction, we currently have one machine with a central reservation system. Users can reserve timeslots for all instances of learning modules from this central location. Every learning context has its own category. Multiple portals can be part of one learning context so that multiple instances of the same learning module can be provided in parallel. Every portal is a website that offers access to the server "behind" it, usually through Java applets. Depending on the learning module this can be a shell on a machine through a SSH terminal emulator applet, a management console on a Cisco router, an editor applet for programming exercises or any other tool that can run in a Web browser.

3.1. Current architecture

In our laboratory setup, independent hardware for different learning modules is used [2, 19]. This has the advantage that changes on one module do not affect most of the others, but this comes at the price of administrative overhead and underutilised hardware. We had about 20 machines dedicated to the e-learning infrastructure, many of them completely idle most of the time—most activity is concentrated on the later part of the semester, so for about four months of a semester the machines are not used at all. Still the machines are powered on most of the time to provide access to all potential users.

Recently, multiple small hardware failures occurred in the laboratory. Because of the amount of machines the administration requires much time. This combination of underutilised hardware and administrative complexity forced us to reassess the current laboratory setup.

3.2. New laboratory setup

As a possible solution we started evaluating virtualisation methods to consolidate our hardware. This will free hardware for parallel installations as reserves in case of failures. Some other advantages are fast recovery as a virtualised machine can be regenerated from a backup within minutes, easy creation of new machines and, better utilisation of hardware. Because we are using Linux almost exclusively we had a large number of virtualisation options. Among the well-known tools are VMWare (commercial) [15], Xen [18], Virtualbox [14], Linux vserver [10], QEmu, OpenVZ and many more. Out of this rather large number of tools we took Xen, Virtualbox and Linux vserver into the closer evaluation. The features these solutions offer comply with the requirements of the targeted laboratory setup. Virtualbox is a full machine virtualiser/emulator. It can run arbitrary operating systems and has some interesting features like a "virtual" graphics card exported over VNC to allow remote access as if it were a local machine. The drawback is that the performance is quite moderate and it is not as convenient to manage. Xen offers "paravirtualisation" where the guest operating system gets adapted to the virtualisation environment. This limits the available operating systems for the guest systems, but as we were already very much focussed on Linux this was no drawback. The advantage is a comparatively good performance, so the amount of hardware needed should be smaller than with other virtualisers. Linux vserver is the most limited of the three, it uses one kernel for all virtualised instances. Within the virtual machines many functions are disabled for security reasons (for example access to the routing table, creation of device nodes). This limits the use of Linux vserver, but the advantage is that all VMs share resources like memory and the overhead is negligible.

Due to the different requirements of the portal and experimentation computers a hybrid approach seems to be advisable. The portal servers would be consolidated onto one Linux vserver instance. This makes sense as they do not use many resources. The Linux vservers offer encapsulation so that each portal can still be managed on its own. However, due to the structure of Linux vserver it now takes only about five minutes to create a new instance. The management is very easy as the host offers direct file-level access to the Linux vservers. This also allows easy backups and maintenance from the host. At the same time resource usage is low, compared to the other virtualisation systems. Each new vserver takes about 50 MB memory when running and idle. Consolidating all existing portals does not cause any performance issues for the single machine they are running on.

For the experimentation computers Linux vserver is not an option. Access to functions like routing tables or kernel modules is needed, so we had to find another solution for those. Xen offered the best compromise of flexibility and ease-of-use. We can consolidate up to 15 virtual machines onto one server; with other hardware we could potentially create even more virtual machines, but there are some limits like the available amount of main memory. If only one or two VMs are active this may be tolerable, but in the rare case that all virtual machines are in use they would all be very slow. Our current server hardware is a dual-processor machine with 2 GB RAM. It is about four times as fast as the previous dedicated machines. Two of those machines provide enough power to provide most of the previous 20 dedicated machines. Installing and managing Xen is not quite as easy as Linux vserver, but with some site-specific documentation and support scripts it is easy enough to work with it.

To facilitate the management we decided to add some monitoring. Using standard SNMP we are able to watch most relevant data points, for example processor load, memory usage, network traffic and whether a certain service is running. For data collection and representation we have evaluated a few monitoring applications, currently Cacti [6] is used and we are testing Nagios [11] in parallel.

With e-mail notifications the administrators can be warned whenever thresholds, like system load or disk space, are exceeded. This allows preemptive intervention instead of reactive administration when things have already failed. Still most management tasks are fully manual.

Figure 4 shows the conceptual design and separation of the virtual machines and physical servers. On the right side the Linux vserver host with the public portals can be seen, with the cold-standby as a shadow below it. On the left side the Xen hosts and their virtual guests are shown. Both Xen and Linux vserver hosts can be managed remotely, but only the latter needs to be exposed to the public Internet.

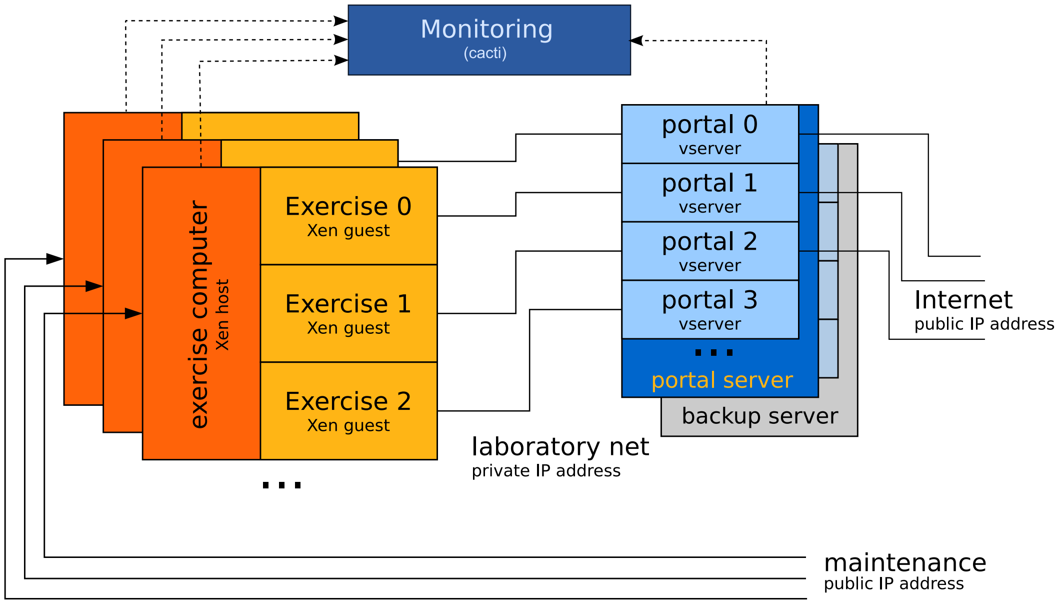

Figure 4. The proposed laboratory architecture.

Some of the existing machines cannot be integrated into the Xen virtualisation at the moment, mostly network- and hardware-specific devices. Still we are able to reduce the number of machines from around 20 to one portal server, two Xen machines and six non-virtualised machines. A backup of the portal and Xen machines is maintained on similar hardware to provide a "manual failover" in case one should fail. In this case the backup machine is booted and the relevant virtual machines are started with a script. This frees a lot of hardware for other purposes and significantly reduces administration time.

4. Conclusions

Two issues of our current e-learning infrastructure have been discussed. The new content formatting tool helps to maintain the sustainability of the course material by supporting an easier migration to new learning management platforms. A simplified document handling makes the creation of learning content more convenient for the learning module author and simplifies subsequent changes to the document structure.

By storing the learning modules in the XML format under a version control system the maintenance overhead is decreased. All changes are now made inside this repository and by using the formatting tool any course module can be retrieved on demand and converted into the desired format. This prevents inconsistencies if the same module is used in several courses.

The migration and virtualisation of our hardware has been a success for us. The necessary hardware has been reduced by more than 50% and the utilisation of hardware could be increased. At the same time we have a much better scalability—the single portal machine still has spare capacity to set up more portals if needed. We have not had any failures to test the failover capability, but we expect that our reaction time will also be much better than the "hours to days" reaction time of the old infrastructure.

The mix of virtualisation methods may be unusual, but it offers us optimal flexibility with only a small documentation overhead. With the help of documentation and support scripts it takes a few minutes to learn how to create a new virtual machine and then only a short time to setup and configure is required. Compared to the old infrastructure this is a huge improvement in productivity. At the same time we use less power and less space. While it is not applicable everywhere, virtualisation offers tools to consolidate underutilised machines and increases the availability at no extra cost apart from the migration itself. Backup and recovery become faster and easier.

References

[1] A. Baier, T. Bernoulli, T. Braun, C. Graf, and U. Ultes-Nitsche. Case study of the usage of an authentication and autorization infrastructure (aai) in an e-learning project. In *Information Security South Africa (ISSA 2006)*, Sandton, South Africa, July 2006.

[2] A. Berqia, A. Diop, and J. Harms. A virtual laboratory for practical exercises. In *Proceedings of International Conference on Engineering Education*, Manchester, UK, August 2002.

[3] Blackboard Inc. WebCT Vista (Blackboard). `http://www.blackboard.com/us/index.Bb`. last visited: December 2007.

[4] T. Braun, M.-A. Steinemann, and A. Weyland. VITELS – an e-learning course on computer networks and distributed systems. *SWITCH journal*, 2:32–35, November 2003.

[5] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau, and J. Cowan. Extensible Markup Language (XML) 1.1, W3C Recommendation. `http://www.w3.org/TR/xml11/`, August 2006.

[6] Cacti. `http://www.cacti.net`. last visited: January 21, 2008.

[7] B. Collins-Sussman, B. W. Fitzpatrick, and C. M. Pilato. *Version Control with Subversion*. O'Reilly Media, Inc., first edition, June 2004.

[8] eduSourceCanada. `http://www.edusource.ca/english/home_eng.html`, 2007.

[9] Edutech. Learning object repository – test results of feasibility study. `http://edutech.ch/lms/2006LOR/index.php`, 2006.

[10] Linux-vserver Virtualization Software. `http://linux-vserver.org/`. last visited: December 6, 2007.

[11] Nagios. `http://www.nagios.org`. last visited: November 30, 2007.

[12] SCORM – Sharable Content Object Reference Model. `http://www.scorm.com`. last visited: December 5, 2007.

[13] SWITCH (The Swiss Education and Research Network). AAI – Authentication and Authorization Infrastructure: System and Interface Specification, 2004.

[14] Virtualbox Virtualization Software. `http://virtualbox.org/`. last visited: December 6, 2007.

[15] Vmware Virtualization Software. `http://vmware.com/`. last visited: December 6, 2007.

[16] A. Weyland and T. Braun. OSLab Module Author Guide. Technical Report IAM-06-004, University of Bern, IAM, July 2006.

[17] M. Wulff and T. Braun. OSLab: An interactive operating system laboratory. *ERCIM News*, 71:46–47, 2007.

[18] Xen Virtualization Software. `http://xen.xensource.com/`. last visited: December 6, 2007.

[19] S. Zimmerli, M.-A. Steinemann, and T. Braun. Resource management portal for laboratories using real devices on the internet. *ACM Computer Communication Review*, 33(3):127–135, July 2003.

Distributed Software Architecture and Applications for Remote Laboratories

Sorin Aurel MORARU
Transilvania University of Brasov,
500024, Politehnicii 1, Brasov - Romania
smoraru@unitbv.ro

Mihai ROMANCA
Transilvania University of Brasov,
500024, Politehnicii 1, Brasov - Romania
romanca@unitbv.ro

Paul BORZA
Transilvania University of Brasov,
500024, Politehnicii 1, Brasov - Romania
paul.borza@siemens.com

Abstract

The paper presents the importance of offering to the users the possibility to do laboratory tests with real data, within remote laboratories using distributed application. The article describes the facilities offered by the Java language in making user interfaces for the remote and virtual laboratories. At the same time, we present some implementations within the transnational project "VetTrend" of the "Leonardo da Vinci" program that have user interfaces based on the Java Applet technology, for accessing the laboratory resources via Internet. Because the connection between the two computers is done through a physical space, on which we may not have any control, whether that connection is through wires or wireless, we must be on guard of any possible identity and information stealing. Also, the security should always be of concern.

1 Introduction

Industrial software applications have an important role in improving very consistently the work efficiency in industry. The effectiveness of industrial electronics is increased by remote control of processes.

We started the research on remote experiments in electric and electronic domain by a large analysis of the state of the art, mainly reflected in the collection of published papers in IEEE Transactions on Industrial Electronics and Transactions on Industrial Informatics, in the period 2005-2007.

The importance of software remote surveillance is provided in papers such as:

-- "Analysis and Software Implementation of a Robust Synchronizing PLL Circuit Based on the pq Theory" [1], where the authors present an analysis and software implementation of a robust synchronizing circuit, designed for use in the controller of active power line conditioners, and

-- "Development of Real-Time Vision-Based Fabric Inspection System" [2], where the authors present an automatic vision-based system for the quality control of web textile fabrics.

In the article "Virtual-Environment Modeling and Correction for Force-Reflecting Teleoperation With Time Delay" [3], the authors consider Virtual environment (VE) as an effective method to deal with time delay in teleoperation, and describe the VE-based teleoperation system developed in their laboratory.

The need for interoperability is prominent in the industrial enterprise environment. Different applications and systems that cover the overall range of the industrial infrastructure from the field to the enterprise level need to interoperate.

In the article “Vertical Integration of Enterprise Industrial Systems Utilizing Web Services" [4], the authors present a distributed system architecture that utilizes dominant state-of-the-art standard technologies in order to address the above issues in an efficient way.

The based remote control scheme for networked control systems is presented from the point of view of the quality-of-service (QoS) in the paper "QoS-Based Remote Control of Networked Control Systems Via Profibus Token Passing Protocol" [5].

But, on a hand, not all the time things work in production environment. In research and development area is important to use many and, sometimes, very sophisticated resources. They are very expensive and difficult to obtain, especially in universities.

On the other hand, there are researches in very dangerous or toxic environments.

From these and other reasons, it is important to put in common resources (accessible from many departments) and to obtain the remote control of them.
We present in our paper an example of developing and operating of such remote laboratories, together with practical issues that we solved in this approach. There are in accordance with the trend in responding to the important constraints of the modern industrial applications of electronics and informatics: **time** (too short), **space** (too few), **users** (too many) and **equipment** (too expensive).

2 Introduction

The remote laboratories present a high interest for the ones that learn, because they have the possibility to experiment on real equipment from home or from work, using equipment located into a university laboratory for example [6], [7]. This facility is important in understanding the processes from the practical point of view. The equipment, often expensive from the laboratories will become accessible to the wide public.
As principle of functionality, electronic driven systems can accept remote commands sent by third party applications, through Internet, as this is an electronic environment. An important issue for the user is what can he obtain from a laboratory, using such a communication system. Obviously he will not be able to access the laboratory as if it was there. Here, the user interface comes in, which is usually represented by a web page. Both the way in which the page is made, and what it will allow the user to do, have a great impact on the learning and understanding of the processes that take place in the real laboratory.
For developing the remote and virtual laboratories user interfaces we use Java. The software development is nowadays one of the most extended domains, and sure, the most dynamic. In such a dynamic environment as IT, time is essential, and this can be notice in the general IT marked trends over the world. Everybody wants better productivity when it comes to developing software, which is basically translated into obtaining similar products over a significantly shorter period of time. The answer to better software development productivity is Java because the many features it encapsulates frees the programmer from a lot of work, when compared, for instance, with C++ [8].
The Java language Sun Microsystems describes Java as a "simple, object-oriented, interpreted, robust, safe, architecture independent, portable, high performance, multithreaded and dynamic" programming language.
Each feature mentioned above is an important part of the Java programming language, as well as a critical request for web programming. The combination between these features makes Java a powerful and useful programming language that supplies the programmer with the tools he needs to easy create powerful programs for the distributed applications environments of today [9].
On the Internet, the Java programs are called *applets*. The applets are Java applications that are embedded into HTML web pages and can be loaded into a Java capable web browser, at a click of the mouse.
Because of its extended portability and easy programming, Java is the optimal choice when developing client applications. Although Java programs are slightly slower than C++ programs and have some disadvantages concerning the use of platform specific features, Java is the preferred programming language for many software developers around the world.
Due to the fact that there exist a lot of operating systems (Windows, Unix, Linux, MacOS, etc) the client applications, if we would use C++, should be written for every platform. Using Java, they must be written only once, on any of the platforms and will be run-able on any other platform without changes. This advantage, along with others offered by Java, clearly places it on the top of client applications programming language [8].

3 Developed applets

3.1 The web-page

All the Java Applets are integrated as objects into web pages (a separate web page for each applet). Because the applet is used only as tool for posting measurement requests and for viewing the measurement results, it does not contain information about how does the experiment take place or about the meaning of the values sent or received. In order to make the applet easy to use by uninitiated students, we provide basic information on how to use the applet and which is the meaning of the values in the experiment in the same web page (which contains the applet).
There are two possibilities to present the descriptive text:
1) On the top of the page, before the applet. In this case, the applet is at the bottom of the page and the user must scroll over the descriptive text in order to reach the applet and make the experiment.
2) After the applet. In this case, the applet is at the top of the page, and the descriptive text after the applet. In this case, the user will read the information only if necessary.
This descriptive text has two meanings:
-- as laboratory guide for first experiments;
-- as help afterwards.
The text placed before the applet is useful in the first stage, because the student can see that there is information about the experiment, but it can be uncomfortable for students that already know how to

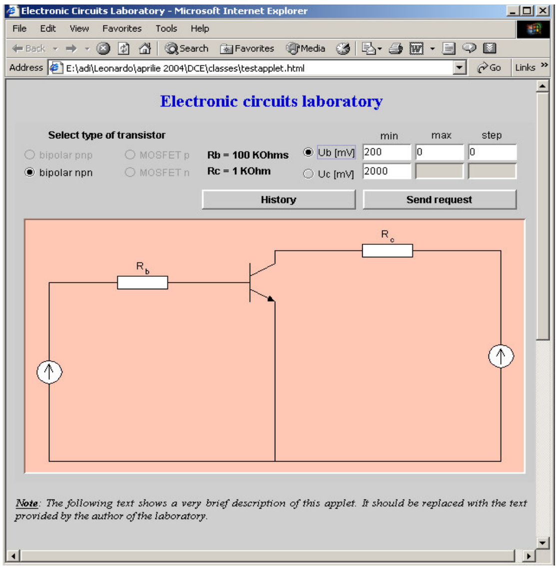

Figure 1: The main view of the Electronic Circuits Laboratory

use the applet, because they have to scroll the text every time they enter the page. The text presented after the applet is better fit for students that know how to use the applet. It can be a problem for students that don't know how to use the applet. They may not see that there is a descriptive text after the applet.

The e-Learning course that corresponds to the laboratory experiment can be accessed very easy from the web page that contains the applet, because there is a link to the course topics page. We have split the process of posting a request and seeing the results into three, separate and asynchronous operations:

- -- posting a request;
- -- viewing a list of posted requests;
- -- viewing the graphics for a certain item of the list.

3.2 Implementations made using the asynchronous mechanism

The web page team has developed a database using SQL Server used as base for the final version of the system. Using this database, we have integrated the functionality for all the applets, corresponding to the following electric and electronic domains: Electronic Circuits, Automated measurements, Electrical Drives, System Theory and Automatic Control, Domotics, Databases with Applications in the Electrical Domain, Electric Circuits, Electrotehnic Materials, Electrical Machines. We will exemplify with the laboratories of Electronic Circuits and Electrotehnic Materials.

The web page that contains the applet of Electronic Circuits Lab presents the descriptive text at the end of the page and the applet at the top, which practically represents the second approach presented at the beginning of chapter 3.2.

The main view (Fig. 1) of the applet is used to post requests. The user can choose which source (*Ub* or *Uc*) will vary it's voltage and the values for the voltages (*min*, *max* and *step*). After pressing "Send request" the applet will send the request to the server and will present the history view, which is also accessible by pressing the "History" button. The type of transistor can be only "*bipolar npn*" because the physical circuit does not support the other types yet.

The history view shows a list with the posted requests. The user can choose a request and click "View the selected request" to view the graphic results for that request. Clicking "Refresh the table" will refresh the information (show the new status from the database).

When the user clicks "View the selected request" the applet gets the result corresponding to the selected request from the database and opens a new window into which a graphic representation is drawn.

The web page that contains this applet of Electrotehnic Materials Lab presents the descriptive text at the top of the page and the applet at the bottom, which practically represents the first approach presented at the beginning of chapter III.A.

The first view (Fig. 2) presented by the applet is used for posting measurement requests. When pressing "Send", a request is posted on the server. When pressing history, the applet shows a table containing all the posted requests, presented in the Fig. 3. When the user presses "View" the applet will show the result, presented in the Fig. 4, for the selected request.

If the "Refresh" button is pressed, the table that contains the list of requests is refreshed so that it shows the current requests configuration.

In Fig. 4 are presented the measured values for the selected request (in the History view). Three types of graphics are available (B(H) presented in the screenshot, B(t) and H(t)).

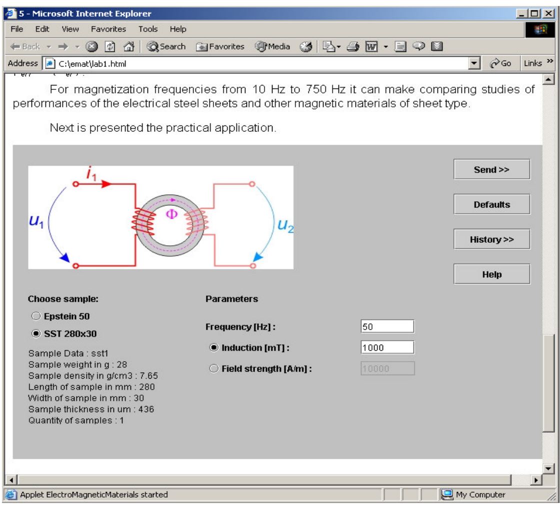

Figure 2: The main view of the Electrotechnic Materials Laboratory

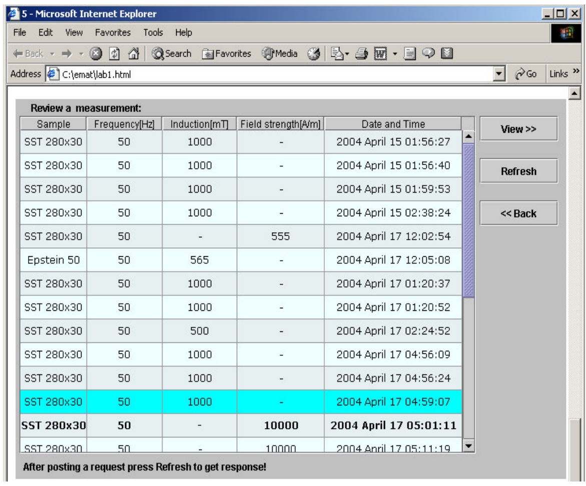

Figure 3: The history view for the electromagnetic materials

The table shows the individual measured values and the textbox in the lower-left corner shows the measured graph points.

The applet checks the values put by the user and does not send the request if the input values are not in the valid range. The user is prompted if he enters invalid values.

The Fig. 2 presents the layout of the main applet view. The picture located at the upper-left corner shows a simplified schema of the measurement equipment. The two radio buttons below the schema can be used to select the type of sample. Below the radio-buttons are printed the specifications for the selected sample.

The user can choose to enter either the value for induction, either for the field strength, by choosing the appropriate radio-button located in the lower center of the view.

The "Help" button will show a HTML page containing the description of the laboratory and documentation on how to use the applet to take experiments.

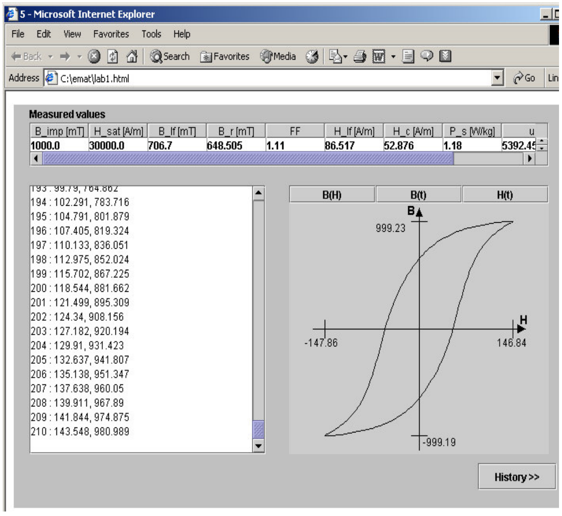

Figure 4: The result view for the electromagnetic materials showing the graph and measured values

By using the view in Fig. 2, a user can see all the requests he made over time. Just as in mail systems, the requests and corresponding responses, which were already viewed, are written normally. The new records are marked with bold. The user can select one of the records and the selected record will be shown with a cyan background. When a record is selected, the view button will become active, otherwise being inactive. The Refresh button will bring the new records from the database.

For knowing which measure was viewed and which not, when pressing the "View" button, the information about the response for the selected request is being retrieved from the database. The application also updates the request status (as being viewed) in the database when making the request.

4 Communication between system levels. Authentication as a main part of the security aspects

4.1 Communication between system levels

Any communication is realized between two or more subjects. Each of them must identify itself to the others, in brief or in details, before being able to participate to that communication [10]. After that, a continuously switch between emitter and receptor roles takes place among the subjects.

In our case we have a communication between the client computer, which makes requests and two servers that respond: the *Learning Management System* (LMS) server and the *VetTrend* (VEL) server.

The communication procedure and the system levels are presented in Fig. 5 and 6.

4.2 The client authentication

In order to maintain a communication with more than one client, the application uses a session mechanism, meaning that it allocates resources (variables, threads, memory etc.) for each one of them, and an identification, which distinguishes one client from the other. Every time a client makes a request, it has to reveal its own session ID, letting the server know its identity and the stage of the communication.

For initiating a communication, the client must send his username and password to the LMS server. The server will verify the existence of this user into the database and depending of the result, will allow the client to connect to the features of the application or not. In case the login succeeds, the server allocates new resources for this new client and sends the ID of this newly created session back to the client. As mentioned, the client will use this session ID every

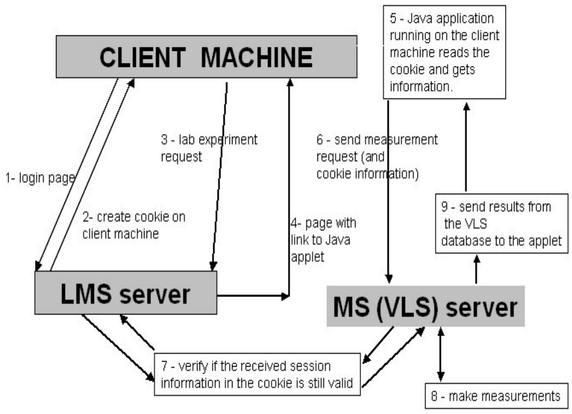

Figure 5: Communication procedure

single time it makes a request to the server, identifying itself among all others connected users.
The use of the pair username / password has another reason too. It allows the access of the client to different features of the application, depending of his role: administrator, teacher, student etc. (Fig. 7).
The VEL server has not a login mechanism of its own, but it will use the login feature of the LMS server. Thus, if a client accesses the VEL server without passing through the checking made by LMS through username/password pair, it will be rejected. The VEL server knows if a client has successfully passed the login, by looking at the session ID provided by the client along with the request.

4.3 The server authentication

Not only must the client identify itself, but also the server. Because the application of VEL is very complex, a part of it is executed on the client machine. For this reason, a Java Applet has been chosen as solution. Within it several computing

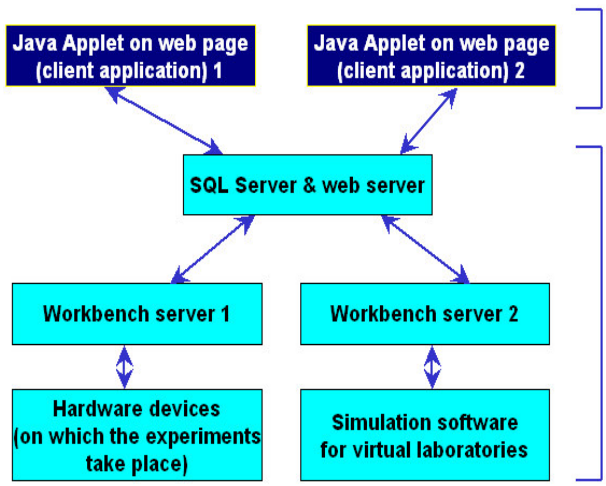

Figure 6: System levels

operations are performed on the client part. In this way, we can reduce the network traffic, because instead of sending multiple requests to the server for different manipulation of data, we can receive the data in the Java Applet client and then all the wanted actions regarding that set of data can be done locally, without any new request to the server.
But this facility offered by the applet raises another issue. The Java Applet, being capable to execute commands on the client machine can executed unwanted commands in order to harm the client's computer.
Because of this potential problem, an applet is not allowed to perform freely, but instead it is subject to a so-called "sandbox", that is a delimited set of permitted actions, anything else outside this sandbox not being available for execution. Among the actions prohibited, are accessing client's files, opening channels of communications to other computer than that where the Applet originated etc.
However, if an application really needs to access some resources on the client machine, such as the file system, the printer etc., there is a mechanism through which this can be accomplished. This mechanism is called "Applet signing". By signing it, an applet requires, at its initialisation, the acceptance from the user to execute actions outside the sandbox [12], [13].
Therefore, when one loads into a browser a Java Applet, he must know who the server is in order to make a clear judgment about if he trusts it or not to let it run commands on his computer. That brings us to the server authentication matter. For this very reason a server must be able to identify itself in order to make the client trust that it is who it claims it is and allowing the clients deciding if they trust it or not. In this way, a third party server cannot claim to be someone else for deceiving the users and running malicious code on their machines.

4.4 Certificates

The first thing in authentication is the creation of a certificate that describes the entity of the claimer. It is made of the main details of the entity: name of person, company name, department name, location, country. This would be the equivalent of an ID of a person in the real world.

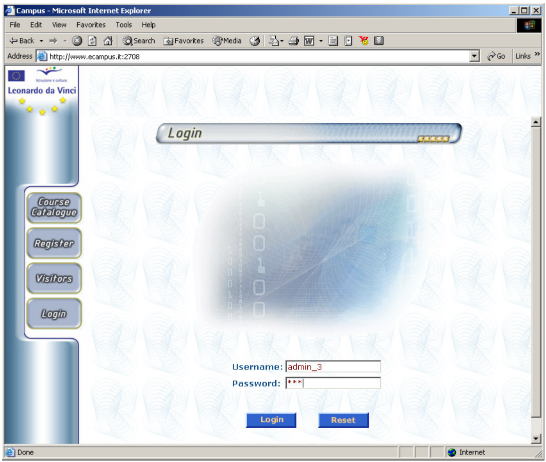

Figure 7: Login page for the LMS application

Now, some authority in the domain must authenticate this certificate, in the same way that in the real world an authority issues a driving license or an ID. In case of digital signing, there are some companies like DeutschePost, Thawte, Verisgn or Ecquifax each one known as Certificate Authority. These companies are entitled to validate those certificates by priory verifying the credentials of the claimer. Then, it applies its own signature over the entity's certificate, thus validating it.

4.5 Packing the Java Applet

The next thing after having the signed certificate, is packaging all the components of the Applet in order to signing it. This is done with the JAR tool.

4.6 Signing the Applet

The third step represents the actual signing of the Applet, now packed into a Java ARchive. The tool used for the operation is *signtool*, provided with every Java Development Kit.
When a user loads a web page that contains a signed applet, a warning is presented to the user (Fig. 8).

Figure 8: Warning of loading a signed applet, which requires additional privileges for the execution of commands outside the sandbox

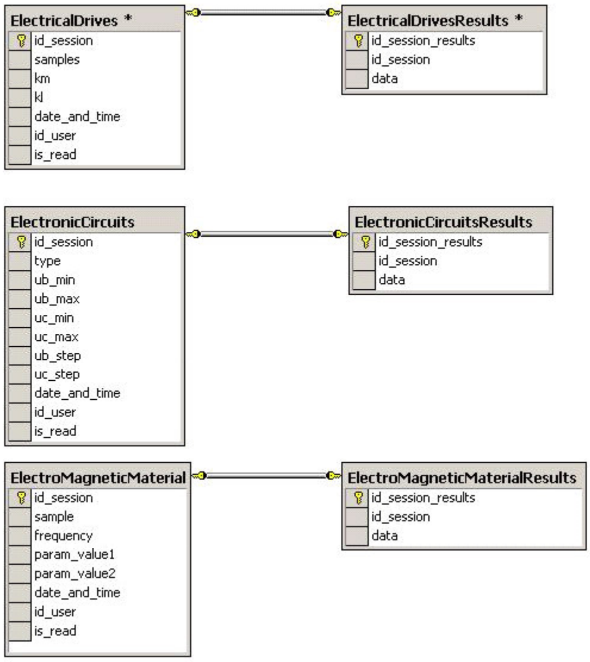

Figure 9: Database structure for three laboratories

The user can see the details about the certificate by clicking "More Details".

5 Database Structure

The fig. 9 illustrates the structure of three tables associated with three of the nine realized remote laboratories.
The tables on the left hold the measurement requests and their structure differs from one laboratory to another.
The tables on the right hold measurement responses and, for each table in the left, there is a corresponding table on the right.
For each record in the left table there is one corresponding record (with the same id and id_session) in the right table for a solved request, and for unsolved requests there is no corresponding record. This record (response) is created by an application, which does the actual data measuring.

6 Conclusions

The use of Java applets as web-based user interface for real laboratories increases the usability of remote instruments. The user capability to understand the phenomena that take place in the real experiments is improved by the use of real data.
Developing intuitive user-friendly interfaces we achieve the goal of making the training material interesting for all the users and transforming the act of learning into a pleasure.

Using the Java technology we provide a good portability on various platforms for the applets, so the users are not restrained to use a specific operating system. Java offers a fast and performing way of developing web-based applications with a wide area of accessibility

References

[1] L. G. B. Barbosa Rolim, D. R. Rodrigues da CostaJr., and M. Aredes, "Analysis and Software Implementation of a Robust Synchronizing PLL Circuit Based on the pq Theory," IEEE Trans. on Industrial Electronics, vol.53, no. 6, pp. 1919-1926, Dec. 2006.

[2] C.-S. Cho, B.-M. Chung, and M.-J. Park, "Development of Real-Time Vision-Based Fabric Inspection System," IEEE Trans. on Industrial Electronics, vol. 52, no. 4, pp. 1073- 1079, Aug. 2005.

[3] Li Huijun and Song Aiguo, "Virtual-Environment Modeling and Correction for Force-Reflecting Teleoperation With Time Delay", IEEE Trans. on Industrial Electronics, vol. 54, no. 2, pp. 1227-1233, Apr. 2007 [3], the authors

[4] A.P., Kalogeras, J.V. Gialelis, C.E. Alexakos, M.J. Georgoudakis, and S.A.Koubias, “Vertical Integration of Enterprise Industrial Systems Utilizing Web Services", IEEE Trans. on Industrial Informatics, vol.2, no. 2, pp. 120-128, May 2006.

[5] Kyung Chang Lee, Suk Lee and Man Hyung Lee, "QoS-Based Remote Control of Networked Control Systems Via Profibus Token Passing Protocol", IEEE Trans. on Industrial Informatics, vol. 1, nr. 3, pp. 183-191, Aug. 2005.

[6] M.Cassini, D. Prattichizzo, A. Vicino, E-Learning by remote laboratories: A new tool for control education, Siena University, Italy, 2002.

[7] J.Trevelyan, Remote Laboratories and Team Skills in Mecatronics, Mecatronics and Machine Vision in Practice 2003 Conference, Western Australia, 2003.

[8] A.Taylor, JDBC Developer’s Resource, Informix Press, 1997.

[9] C.Horstmann, G. Cornell, Core Java 2 vol. I and II, Sun Microsystems Press, 2001.

[10] S.A.Moraru, A. Pelcz, C. Bujdei, L. Perniu, Web-oriented Applications of Databases Used in Electrical Domain, The 14th International Scientific and Applied Science Conference Electronics’2005, September 21-23, 2005, Sozopol, Bulgaria, book 3, ISBN 954-438-519-3.

[11] S.A.Moraru, I. Diaconu, A. Pelcz, S. Leahu, The Database and Architecture of Virtual and Remote Laboratories, 1st International Conference on Informatics and Control, Automation and Robotics, p. 38-42, ISBN 972-8865-14-7, august 2004, Setubal - Portugal.

[12] http://ecommerce.mit.edu/class/private/crypto_summary.html

[13] http://java.sun.com/developer/onlineTraining/Security/Fundamentals/Security.html

[14] http://www.openssl.org

Tele-Lab IT-Security: an architecture for an online virtual IT security lab

Christian Willems
Hasso-Plattner-Institut für Softwaresystemtechnik
Prof.-Dr.-Helmert-Straße 2–3
Potsdam, Germany
christian.willems@hpi.uni-potsdam.de

Abstract

Recently, Awareness Creation in terms of IT security has become a big thing – not only for enterprises. Campaigns for pupils try to highlight the importance of IT security even in the user's early years. Common practices in security education – as seen in computer science courses at universities – mainly consist of literature and lecturing. In the best case, the teaching facility offers practical courses in a dedicated isolated computer lab. Additionally, there are some more or less interactive e-learning applications around.

Existing offers can do nothing more than impart theoretical knowledge or basic information. They all lack of possibilities to provide practical experience with security software or even hacker tools in a realistic environment. The only exceptions are the expensive and hard-to-maintain dedicated computer security labs. Those can only be provided by very few organisations.

Tele-Lab IT-Security was designed to offer hands-on experience exercises in IT security without the need of additional hardware or maintenance expenses. The existing implementation of Tele-Lab even provides access to the learning environment over the Internet – and thus can be used anytime and anywhere. The present paper describes the extended architecure on which the current version of the Tele-Lab server is built.

1. Introduction

The increasing propagation of complex IT systems and rapid growth of the internet more and more attracts notice to the importance of IT security issues. The limits of technical security solutions are fixed by the lacking awareness of computer users, caused by laziness, inattentiveness and missing education. In the context of awareness creation IT security training becomes a topic of strong interest – as well as for companies as for indiviuals.

Traditional techniques of teaching (i.e. lectures or literature) have turned out to be not suitable for security training, because the trainee cannot apply the principles from the academic approach to a realistic environment within the class. In security training, gaining practical experience through exercises is indispensable for consolidating the knowledge [2].

Precisely the allocation of an environment for these practical exercises pose a challenge for research and development. That is, because students need privileged access rights (root/administrator-account) on the training system to perform most of the imaginable security exercises. With these privileges, students can easily destroy a training system or even use it to attack other computers.

The classical approach is to provide a dedicated computer lab for security training. Such labs bare different backdraws: they are immobile, expensive to purchase and maintain and must be isolated from all networks.

Teleteaching approaches for security education are multimedia courseware or demonstration software, which do not offer practical exercises. In simulation systems users have kind of hands on experience, but a simulator doesn't behave like a realistic environment and simulation of complex systems is very difficult. Those approaches to security education will be presented more precise in section 2.

The Tele-Lab project (at first proposed in [10]) provides a novel e-learning system for practical security training in the WWW and inherits all characteristics from offline security labs. The Tele-Lab server basically consists of a web based tutoring system and a training environment built of virtual machines. The tutoring system offers three kinds of content: information chapters, introductions to security- and hacker-tools and finally definitions of practical exercises. Students perform those exercises on virtual machines (vm) on the server, which they use via remote desktop access. A virtual machine is a software system, that provides a runtime environment for operating systems. Such software-emulated computer systems allow easy deployment and recovery in case of failure.

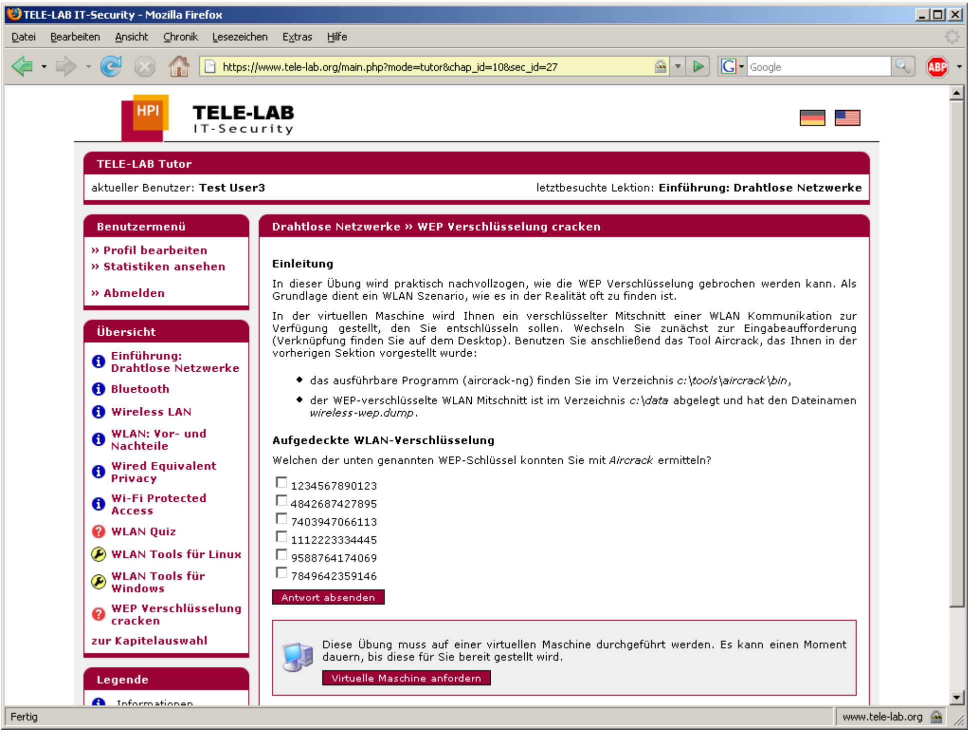

Figure 1. Screenshot: exercise on Wireless LAN in the Tele-Lab tutor

A learning unit on e.g. "wireless networks" introduces to different WiFi technologies like Wireless LAN or Bluetooth, explains the functionality of mechanisms and protocols for wireless security and highlights weaknesses which lead to security problems. Thereafter, the tutoring system presents wireless tools for Windows and Linux like "Kismet" or the "Aircrack Suite". The chapter concludes with an exercise, where the student is asked to reveal a WEP encryption key from a wireless traffic dump file using "aircrack".

For that exercise, the student requests a virtual machine (here: Linux or Windows). If there is a free vm on the server, the student will be assigned to that vm and a remote desktop session will be started in an applet window. After performing the exercise (cracking the wireless dump), the student must enter the revealed WEP key in the tutor and prove, that he solved the exercise (see figure 1). The vm will be reclaimed and restored to its original state automatically.

Currently, there are also chapters on access control, authentication, encryption, intrusion detection, malicious software, man-in-the-middle-attacks, packet sniffing, port scanning and secure email. Additional chapters can be authored and integrated easily.

The architecture for providing this learning environment is described in section 3. The paper will be concluded with a summary of the results and an outlook on future work in section 4.

2 Related work

Related work in security education mainly includes Web-based training and multimedia courseware, demonstration software, simulation systems, and dedicated computer laboratories for security experiments. Recently, effords in compiling practical computer science courses using virtual machine technology have turned out some similarities to the Tele-Lab approach.

2.1 Web-based training and multimedia e-lecturing

Web-based training (WBT) means e-learning application that present learning units via WWW in the user's browser. Those learning units usually consist of text, images and custom made animations. Extended to multimedia courseware, WBT also offers *real* multimedia content, i.e. audio and video. WBT courses usually are digitized lectures and demonstrations. A modern e-lecturing system is e.g. *tele-TASK* [15]. It is a state-of-the-art streaming system to create online lectures and seminars for teaching IT security. WBT and e-lectures in nature support e-learning on the Web, but they offers no hands-on experiences to learners at all.

2.2 Demonstration software

Demonstration software programs such as *Cryptool* [7] and *CAP ("Cryptographic Analysis Programs")*

[16] are educational suits for learning about cryptography and cryptanalysis. They provide more interactivity for students to play with algorithms and therefore have more practical features than multimedia courseware. However, because demonstration software is mainly used to learn academic cryptographic algorithms, practical network security tools or everyday environments are seldom involved in learning of this kind.

2.3 Simulation systems

Simulation systems are normally used to train students in specific IT security subjects. In [14] and [20], the authors describe, how to familiarize students with intrusion detection by creating audit files with information on user activities. They ask students to detect and resolve an intrusion problem. *CyberCIEGE* [11] is a simulation game in which players construct computer networks and make decisions to protect valuable assets from attacks. Simulation systems offer students chances to perform operations for accomplishing "real" tasks such as identifying intrusion and recovering or cleaning systems. Nonetheless, those security operations are not really performed but simulated in an abstract environment. In fact, real computer systems can be modeled by such simulations only to a very limited degree. Simulation systems increase interactivity to some degree but still offer students no chance to apply real world tools and see what's going on in practice.

2.4 Dedicated computer laboratories

Dedicated computer laboratories for IT security have been created in many universities, e.g. described in [18] or [13]. Security experiments or exercises are usually arranged on the dedicated computer networks. Compared to other approaches, dedicated computer laboratories are ideal environments for practical security teaching because security exercises are performed by application of production software in real systems. However, practical education by laboratory measures normally results in high costs. Dedicated networks require expensive hardware/software investments and intensive efforts to create, configure, and maintain laboratory environments as well as to prepare, supervise, and evaluate exercises. On the other hand, most security exercises require system level access to the operating system. This introduces the risk of misuse and inconvenience of administration. For security reason, dedicated networks are normally operated on isolated networks, which implies that such security laboratories pitifully fail to benefit a wider range of learners outside campus.

2.5 Virtual machine based computer courses

Instead of dedicated laboratories, at some universities virtual machines have been established for courses with special requirements to hard- or software configuration. In [4], the author describes the utilization of UML[1]-based virtual machines for a course in operation systems maintenance, [3] proposes the use of VMware Workstation[2] for courses on network administration, security and database maintenance. The virtual machines are mainly used for providing an environment, where students are allowed to arbitrary configure an operation system and different applications. The virual machines must be compiled customly for every course and somehow distributed to the students. Both authors confirm the opinion of the Tele-Lab research group refering to the virtual machine technology as a cost efficient replacement for physical dedicated labs.
Existing work above indicates the value of a project like Tele-Lab, integrate practical security exercises into e-learning and degrade the costs of laboratory resources.

3 Architecture for the Tele-Lab server

The architecture presented in the paper at hand mainly consists of four components (see figure 2):

1. *Portal* and *tutoring environment* consist of dynamic webpages delivered by an Apache webserver. This is what was described as a WBT application above. Additionally, the tutoring environment provides user controls to require a virtual machine for exercising and initiates remote desktop sessions.

2. The *Virtual machine pool* is a collection of prepared virtual machines primarily managed by an arbitrary virtual machine monitor (see section 3.1 for details).

3. The *database* contains all user profile data, content data and persistent information on virtual machine states.

4. For proxying the remote desktop connections, the server is equipped with a free implementation of the *NX server* (see section 3.2).

5. The central *Tele-Lab control server* (also described in section 3.1) handles control of the virtual machines' state and the remote desktop connections.

[1] User-mode Linux is a Linux kernel for the user-space, allows the operation of virtual machines

[2] A host-based virtual machine monitor application, see `http://www.vmware.com/products/ws`

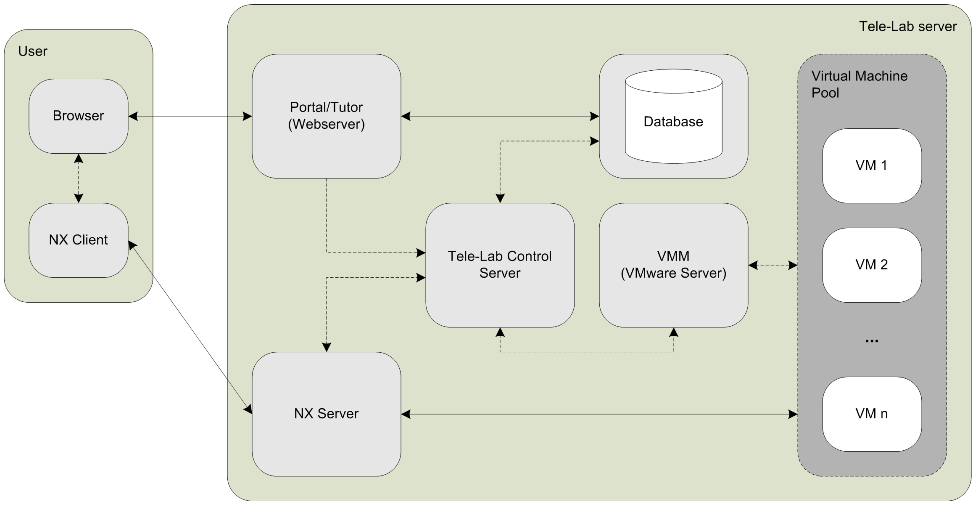

Figure 2. Overview: architecture of the Tele-Lab server

Before describing the important components in detail, the paper will motivate the objectives of the design process for this architecture. It was generally driven by the following requirements that arose from the analysis of previous versions of Tele-Lab such as *Tele-Lab on CD* [9] or the existing prototype of the Tele-Lab server [8].

Handy system usage showed up to be important because all former implementations of Tele-Lab was packed with the (partly deprecated) remainings from the very first prototypes user interface – where Tele-Lab was ment to be an offline/single-user system. Another significant backdraw of the existing UI was the implementation of the Remote Desktop Access as a VNC-applet *inside* the tutoring system. This resulted in a very small remote desktop and wired up situations like *"a browser an a desktop in a browser"*. Design and implementation of the dynamic webpages will not be within the scope of this paper.

Also important was the objective to gain *support for different operating systems inside the virtual machines*. The prototype of the server uses User-mode Linux [5] for the virtual machine pool for reasons of performance, security and availability of alternatives. UML can naturally only support Linux as guest-os inside its virtual machines. But Tele-Lab also wants to address people as users that are bound to Windows systems. A comparison of the available virtualization applications showed, that the VMware products and *Xen* [1] in combination with *Intel's VT* (hardware-aided virtualization [17]) can host both Linux and Windows having Linux as os on the physical host. As it is to remark that Windows guest support is still only experimental in Xen [6], the free *VMware Server*[3] is chosen for the virtual machine pool in this architecture.

Comparing the different virtualization solutions on the market also turned out that vendors and open source projects continously enhance their software suites: more features, more guest os support, better performance. So *being as independent as possible from a concrete virtual machine monitor* should be the next objective. Therefore this paper defines possible states of the virtual machines and proposes an interface with functions to trigger the appropriate transitions for vm control.

Of course, *security and reliability* are important issues for an e-learning system on IT security. Facing the facts: the Tele-Lab server provides remote administrator access to virtual machines on the Internet. This concept rises lots of – partly vertical – security objectives:

- protect the Tele-Lab server from attacks from the internet,
- protect the Tele-Lab server from attacks from malicious vm users,
- protect the internet world and own network against malicious vm users and
- protect vm users against other malicious vm users.

The implementation of the described architecture must prevent attacks from the internet on the web interface (server, scripting interpreter and scripts) and the remote desktop access service. Attacks from inside a virtual machine mainly object the virtual machine monitor: attacks on the hypervisor itself or the attempt to execute own code on the physical machine – bypassing the vmm. Different solutions for some concrete security issues will be described in sections 3.1 and 3.2. Please note that the protection against cross site scripting, code insertion or SQL injection through the user interface and security critical system updates have been considered to be an issue in implementation and administration, but will not be covered in detail here.

[3] see `http://www.vmware.com/products/server`

3.1 Virtual machine management and the generic vmm interface

The virtual machine pool consists of a number of virtual machines provided by the VMware server. Those virtual machines are run from different cloneable os images, each image configured as user vm or victim vm for the one or more exercises. The vm pool is maintained by the Tele-Lab Control Server, a service on the physical host.

Virtual machine states Within Tele-Lab, virtual machines can have a set of different states. Those trigger associated actions in the control monitor – which makes up the virtual machine management. The defined states are:

- *down* – the vm is not running, virtual machine image files exist, the vm is registered with the vmm and could be booted
- *ready* – the vm was started and is running, vm has networking up and a working ip
- *assigned* – the vm has been requested for an exercise and is assigned to a user
- *crashed* – the vm has crashed due to misuse or other guest system errors, transitions to *recovering*
- *recovering* – the vm is being restored to original state and rebooted

Virtual machine control The control server for virtual machine management has been implemented as a java application, that provides a webservice running in an *Apache Axis2* webservice engine. Currently, the services are only accessed by scripts running on the Tele-Lab's webserver. For security reasons, the Axis2 only accepts connection requests from localhost via the loopback interface. At the moment, the control server provides three webservices:

- *VM request* – when a user requires a vm using the HTML control in the tutoring interface, the webserver calls a service to get a vm *assigned*. If there is a free (*ready*) machine, the control server assigns that vm to the user and prepares the remote desktop connection (see section 3.2).
- *Change virtual machine pool* – two more webservices enable Tele-Lab administrators to *start* additional or *stop* unused machines from a web-based admin interface. This changes the state of virtual machines between *down* and *ready*.

More webservices may come, when clustering and communication between nodes will be implemented (see section 4).

The other purpose of the control server is to periodically check all virtual machines' states and eventually trigger actions:

- If a vm's state is *ready*, everything should be in place. In a longer interval, the control server checks, if the vm is still answering (heartbeat ping) and has networking up. If not, the vm is recovered.
- Should a vm be right *assigned* to a user, the control server checks, if the remote desktop connection is still alive. If not, the vm is recovered, the connection is being cleaned up.
- A *recovering* vm is being restored to its original state. In this implementation, the VMware Server make recovery an easy task. The virtual machines have been configured to have *non-persistent virtual hardisks*. This means, changes to a vm during a session are cleaned up whenever the machine stops. Recovery becomes "powering off" and rebooting a vm. A *recovering* machine becomes *ready*, when the boot sequence is finished – which is periodically checked by the control monitor.

VMM interface Speaking of starting, stopping or powering off virtual machines, sending heartbeat pings or checking vm's networking means using any of the virtual machine monitor's interfaces. For automatic control purposes, this can be either a command line interface or some API. All necessary calls to the vmm are encapsuled within a class, that implements the interface `VMControl`, which defines the methods `startVM()`, `stopVM()`, `recoverVM()`, `assignVM()`, `isAlive()` and `isConnected()`.
Currently, the interface is implemented in the `VMwareControl` class, which interacts with the CLI of the VMware Server. For example, a call to `startVM()` causes the `VMwareControl` object to execute `vmware-cmd /path/to/vm/config.vmx start`.
Like this, every method defined in the interface is mapped on database manipulation and CLI execution. Exceptions are `assignVM()` and `isConnected()`, which both also call functions for RDA connection control (see section 3.2).
When setting up the Tele-Lab server with a new virtualization software suite, a new class implementing the `VMControl` interface must be implemented.

3.2 Remote Desktop Access

Classic remote desktop connection environments are based on VNC (Virtual Network Computing), Windows RDP (Remote Desktop Protocol), X11 or terminal server solutions. The prototype of Tele-Lab server

used VNC because of its adaptive bandwidth and the integrated Java-applet client. Though, the choice of VNC has different backdraws. First of all, there is no free available proxy solution for VNC, which causes the need for dynamic reconfiguration of the firewall in the existing prototype. Second is the weak authentication in the VNC protocol which causes possible unauthorized access or connection hijacking. Third is the rather poor performance of VNC connections. Remote desktops provided via VNC usually “feel” slow and sluggish.
The descriped architecture proposes NX[4] for the remote desktop connections. NX infrastructure and protocol offer features and performance that overcome the above described problems of VNC.

NX features [12] NX is a client/proxy remote desktop suite which implements a custom NX protocol and a set of very efficient compression algorithms. A NX remote desktop connection consumes about 40 kbit/s of bandwidth. The NX server is implemented as a proxy and can forward local and remote X sessions as well as remote RDP or VNC sessions. The NX protocol uses SSH for authentication and SSL for optional connection encryption.
The listed features bring everything needed for the Tele-Lab server's architecture. An RDA connection to a Linux machines tunnels the X server of a vm to the user, a remote desktop session on a Windows XP machine uses RDP via NX. The integration of SSH and SSL guarantees enhanced security.
Besides the official implementation of the NX server from NoMachine, there are two free variants of the server: the *freeNX*[5] project and the *2X terminalserver*[6]. The implementation of this architecture uses the 2X server.

NX server integration As already mentioned when describing the Tele-Lab control server in section 3.1, the remote desktop connection management is strongly connected to the virtual machine control. NX allows the configuration of remote desktop connections using `nxs` files (NX session). Those XML session files contain all necessary connection information as key-value pairs: server host, user credentials, connection parameters (screen size etc.), connection forwarding parameters and the SSH key.
NX session files are created by the vm control server, when a virtual machine are assigned to users. The files are stored into the document root of the Tele-Lab webserver. For security reasons, file names of the session files are scrambled via *MD5*. Both, filename and the session file itself are only transferred over HTTPS. When the session file has been established sucessfully, the assigned user must be enabled for NX. Therefore, the command line interface of the NX server is used. The control server executes `nxserver --userenable the_username`.
As this code example exposes, new users of the Tele-Lab server must also be registered to the NX server. This and other administration funtionality is also implemented in the Tele-Lab control server. Further details and features of the admin interface of the Tele-Lab server cannot be covered by this paper for reasons of extensiveness.
For assigned virtual machines, the control server checks, if the connection is still alive. Execution of `nxserver --list` returns a list of active sessions. The detection of an assigned vm with inactive connection triggers the immediate deletion of the corresponding generated NX session file.

NX client integration There is a reliable working beta-version of the NX client as Java-applet available from NoMachine. With the so called `NX WebCompanion` the remote desktop can be integrated into the web-based tutor environment seamlessly. As described above, connection data is stored in an NX session file on the webserver, when a vm is assigned. The tutor script opens a popup browser window with the NX client applet and the URL of the session file as applet parameter. The applet establishs the connection (performs SSH authentications, SSL handshake and negotiation of connection parameters) and starts the remote desktop session.
For security reasions, users are not allowed to copy and paste between the remote desktop and their local desktop.
When the NX connection is quit by the user or interrupted otherwise, or if the virtual machine crashes during a session, vm and NX connection are rolled back to the original states.

3.3 More implementation details

The realisation of the proposed architecture consists of many more components than the above described. Design and implementation of the virtual machine management including the encapsulation of the virtualization solution and the integration of the tutoring environment with the virtual laboratory using the NX technology are just pointed out as the major issues.
Other important implementation aspects have been:

- design and realisation of portal and tutor user interfaces
- database design and optimization
- implementation of administration interface and functionality
- host system setup and configuration

[4]Nomachine Inc. provides commercial NX server but free libraries and client, see `http://www.nomachine.com`
[5]see `http://freenx.berlios.de`
[6]see `http://www.2x.com/terminalserver`

- application and service configuration
- host networking configuration (physical and virtual network interfaces)
- complex host firewall configuration

Every single of the listed tasks poses little challenges described in detail by [19].

4 Conclusions and future work

The present paper describes an architecture for a comprehensive online e-learning solution for IT security. The proposal integrates a web-based tutoring application with a secure virtual laboratory environment. The virtual computer lab consists of various virtual machines. It is accessed via the internet using remote desktop connections. The architetcure has been implemented successfully, content from previous Tele-Lab revisions is already integrated for several chapters.
Future work will focus on different aspects: concerning technology, a second implementation of the vmm interface (e.g. `XenControl`) will be implemented. For better performance (providing a larger number of virtual machines simultanously) implementing clustering Tele-Lab servers could be a needful enhancement for the architecture. Other possible technology upgrades focus on collaborative learning: community tools, messenging and chat or even remote desktop assistance are considered to be integrated.
The second part of future work is the generation of content: implementation of learning units and exercise scenarios, compilation of courses for different learning levels and the identification of future topics.
Third and last aspect for the future is the evaluation of the system as e-learning tool: is learning (are awareness creation campaings) with Tele-Lab IT-Security more efficient than with classical teaching methods? Currently, different high schools over germany test and evaluate a Tele-Lab course with lectures for scholars of different age.

References

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *SOSP 03: Proceedings of the 19th ACM symposium on Operation system principles*, pages 164–177, New York, USA, 2003. ACM Press.

[2] M. Bishop. Education in information security. *IEEE Concurrency*, 8(4):4–8, 2000.

[3] W. I. Bullers, S. Burd, and A. F. Seazzu. Virtual machines – an idea whose time has returned: application to network, security, and database courses. In *SIGCSE '06: Proceedings of the 37th SIGCSE technical symposium on Computer science education*, pages 102–106, New York, NY, USA, 2006. ACM.

[4] R. Davoli. Teaching operating systems administration with user mode linux. In *ITiCSE '04: Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*, pages 112–116, New York, NY, USA, 2004. ACM.

[5] J. Dike. User-mode linux. In *Proceedings of the 5th Annual Linux Showcase & Conference*, Oakland, California, USA, 2001.

[6] Y. Dong, S. Li, A. Mallick, J. Nakajima, K. Tian, X. Xu, F. Yang, and W. Yu. Extending Xen with Intel Virtualization Technology. *Intel Technology Journal*, 10:193–204, 2006.

[7] B. Esslinger. Cryptool – spielerischer Einstieg in klassische und moderne Kryptographie: neue Version – fundierte Awareness in Deutsch und Englisch. *Datenschutz und Datensicherheit*, 26(10), 2002.

[8] J. Hu, D. Cordel, and C. Meinel. A Virtual Machine Architecture for Creating IT-Security Laboratories. Technical report, Hasso-Plattner-Insitut, 2006.

[9] J. Hu and C. Meinel. Tele-Lab IT-Security on CD: Portable, reliable and safe IT security training. *Computers & Security*, 23:282–289, 2004.

[10] J. Hu, M. Schmitt, C. Willems, and C. Meinel. A tutoring system for IT-Security. In *Proceedings of the 3rd World Conference in Information Security Education*, pages 51–60, Monterey, USA, 2003.

[11] C. E. Irvine and M. F. Thompson. Expressing an information security policy within a security simulation game. In *Proceedings of the Sixth Workshop on Education in Computer Security*, pages 43–49, Monterey, USA, 2004.

[12] G. Pinzari. Introduction to NX Technology. Technical report, Nomachine Inc., 2003.

[13] D. Ragsdale, S. Lathorp, and R. Dodge. Enhancing information warfare education through the use of virtual and isolated networks. *Journal of Information Warfare*, 2:53–65, 2003.

[14] N. Rowe and S. Schiavo. An intelligent tutor for intrusion detection on computer systems. *Computers and Education*, 31:395–404, 1998.

[15] V. Schillings and C. Meinel. Tele-TASK – teleteaching anywhere solution kit. In *Proceedings of ACM SIGUCCS*, Providence, USA, 2002.

[16] R. Spillman. CAP: A software tool for teaching classical cryptology. In *Proceedings of the 6th National Colloquium on Information System Security Education*, Redmond, Washington, USA, 2002.

[17] R. Uhlig, G. Neiger, D. Rodgers, A. L. Santoni, F. C. Martins, A. V. Anderson, S. M. Bennett, A. Kagi, F. H. Leung, and L. Smith. Intel Virtualization Technology. *IEEE Computer*, 5:48–56, 2005.

[18] G. Vigna. Teaching hands-on network security: Testbeds and live exercises. *Journal of Information Warfare*, 2:8–24, 2003.

[19] C. Willems. Eine erweiterte Architektur für den Tele-Lab Server: Implementierung eines generischen Interfaces für Virtual Machine Monitors. Master's thesis, Universität Trier, 2006.

[20] C. Woo, J. Choi, and M. Evens. Web-based ITS for training system managers on the computer intrusion. In *Proceedings of the 6th International conference on Intelligent Tutoring Systems*, Biarritz, France and San Sebastian, Spain, 2002.

NeOS: Neuchâtel Online System*

Heiko Sturzrehm, Frederic Aubert, Peter Kropf and Randoald Corfu
Department of Computer Science
University of Neuchâtel
CH-2000, Neuchâtel
Switzerland
{heiko.sturzrehm, frederic.aubert, peter.kropf, randoald.corfu}@unine.ch

Abstract

Most online courses include some form of tests to evaluate the student's performance. Some courses, in particular computer sciences courses, require hands-on training experiences. The NeOS Framework *presented in this paper proposes an e-Learning tool developed supporting in particular hands-on training through programming exercises. The user locally develops his code in an editor-like user interface and then tests his source code on a well secured remote machine. In the case of programming exercises, the* NeOS Framework *supports computer aided evaluation as well as the cross check of solutions submitted. Furthermore, it offers several security measures at the system level by securing in particular the lab machines from being affected by attacks from the outside, and at the usage level by reducing the cheating possibilities.*

1 Introduction

Due to the sustained advances in computer and networking technologies, Internet services and networked applications have become widely available and largely deployed. Leveraging from these developments, e-Learning has become more and more a focus in modern learning methods. It is no more necessary for students and teachers to stay at the same place at the same time, thus giving raise to the virtual classrooms which may additionally be customized to individual needs and requirements. The world wide knowledge is growing everyday and so everyone is now in a lifelong learning process to stay up to date. Virtual classrooms and e-learning in general heavily support this trend.
Since the year 2000 the Swiss Virtual Campus [1] initiative promotes learning over the Internet at the Swiss Institutions of Higher Education. At the moment there are 82 courses online, covering a wide spectrum of disciplines. Among these courses is the Operating System Laboratory, *OSLab* [2] shortly presented in this paper.
Several companies are offering a wide range of e-Learning platforms which support online learning. The WebCT platform (Web Course Tools) [3] from Blackboard has been chosen for the Swiss Virtual Campus (SVC) online courses. In order to simplify the access to the SVC courses, a unique nation-wide user management has been deployed. Indeed, most Swiss universities are member of the SWITCHaai [4], a Swiss wide Authentication and Authorization Infrastructure. This Federated Identity Management enables course providers to use identity information across security domains, e. g. between individual universities (Fig. 1). Furthermore, it deals with issues such as interoperability, liability, security, privacy and trust. In fact with this large scale identity management it is possible to allow students to attend SVC courses at any university in Switzerland.

1.1 OSLab

The Operating System Laboratory, *OSLab* is an online course teaching students the principles of computer operating systems using a constructionist and problem-oriented learning approach. *OSLab* focuses on hands-on training experience of the students and is designed for university undergraduate students. It can be used as an integral coursework or as a complement to traditional lectures. The course shows a modular structure where each module covers one particular topic in a self-contained manner. The tutor responsible for the course can select modules according to the particular needs. Furthermore, the flexible structure easily allows adding new modules to the course. Prior to the *OSLab* we have developed a similar e-Learning course called *VITELS* (Virtual Internet and Telecommunications Laboratory of Switzerland) [5]. Experience with the use of *VITELS* revealed the need for a unified architecture for the hands-on assignments in the course. As it is proper to courses in computer science, as-

*Part of the *OSLab* which is a joined project of University of Bern, University of Fribourg and University of Neuchâtel founded by Swiss Virtual Campus (CVS-IP-4-019).

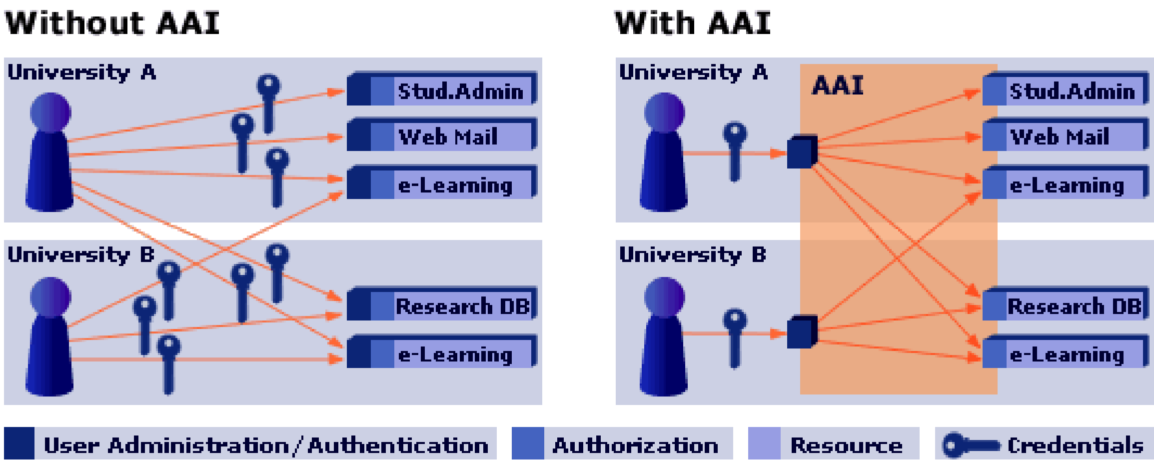

Figure 1. Features of SWITCHaai taken from [4]

signments typically include programming tasks or solutions obtained from executing programs. Solutions to assignments can thus be well-defined which eases automatic evaluating as opposed to the evaluation of free-text solutions needing interpretation for assessing them. This particularity has been taken into account. Therefore, the following requirements have been defined within the *OSLab* project for the development of the *NeOS* (Neuchâtel Online System) hands-on framework:

Compatibility with WebCT / *VITELS* course. Integration into the used course work architecture and existing course.

Authenticy. Decreasing the possibilities to cheat.

Computer aided evaluation. Automated evaluation of hands-on tasks in a way the tutor only needs to verify the (automated) evaluation results.

Flexibility. Platform independence of the evaluation tool and generic customizable deployment

Ease of evaluation. Support for a quick cross check of the solutions.

2 Design and Implementation

The presentation of design and implementation issues of the *NeOS* hands-on framework follows the order of the requirements list shown above. First, we describe the architecture compatible with the *VITELS* course and WebCT. Then, our authenticity solution is presented, and the computer aided evaluation is specified. At the end of this section, we finally present the implementation of the *NeOS* tools. The framework consists of three principal components. The *NeOS Frontend Applet* and the *NeOS Backend Server* are part of the user interface, and the *NeOS Tutor Tool* provides the interface for evaluating the assignments.

2.1 Architecture

The *VITELS* course as well as the *OSLab* course development is a collaborative effort including several partners. The hands-on assignment environment had been individually developed by the partners in the case of *VITELS*. As a consequence, a great variety of technologies had been used for the hands-on tasks. Some modules of the course used Java applets, others had the need of a direct connection to the lab computers only accessible with a Java implemented SSH console, and still others used Java applications running with the Java Web Start utility. However, all have in common the deployment on an Apache Tomcat Web server [6] which used a Shibboleth [7] called plug-in to support the authentication architecture of SWITCHaai.

When planning the new *OSLab* modules, we realized that, in this course, most of the hands-on assignments would include programming tasks which could not be supported with the existing implementations. On one hand, there are simulators which need some source code as input. However, the required simulators are commonly to large to be loaded via the Internet. On the other hand, students shall develop Java programs to be tested on the lab machines. As these machines are firewall protected, the students would have to upload them through a secure SSH connection. While this is certainly a viable solution, it does not prevent students from uploading and executing malicious code.

Taking into account the requirements discussed above, two new components have been added to the existing system architecture as shown in 2: the *NeOS Frontend Applet* and the *NeOS Backend Server*. Both components are programmed in Java [9]. The *NeOS Frontend Applet* serves as as frontend for the students and is deployed on the Web server already used in *VITELS*. In this applet the students are able to program and to submit their program to the *NeOS Backend Server*, which is located on one of the lab machines. On this com-

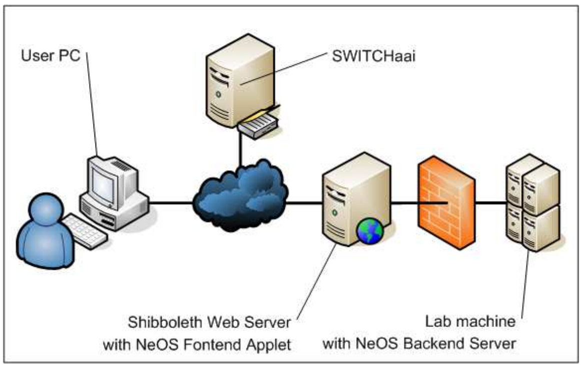

Figure 2. Architecture for NeOS

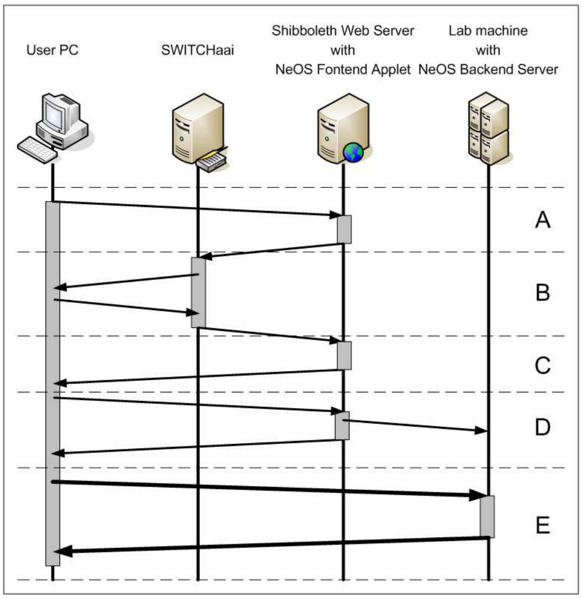

Figure 3. Communication Protocol

puter the (Java) source code of the student will be executed either in a sandbox, or a simulator can be executed with its source code.

The communication protocol used in this architecture is depicted in Figure 3. First the user has to access the webpage on the Shibboleth Web Server where the *NeOS Frontend Applet* is stored (Fig. 3, A). If he is not authenticated, he will be rerouted to the SWITCHaai to do so (Fig. 3, B). With an successful authentication he will then be enabled to download the *NeOS Frontend Applet* into his browser and execute it there (Fig. 3, C). When the applet initializes itself, it establishes an SSH connection via the Shibboleth Web Server to the *NeOS Backend Server* (Fig. 3, D) running on the lab machine. This step concludes the setup phase and the user can now solve his task in the *NeOS Frontend Applet*. If the student wants to test his solution, the *NeOS Frontend Applet* will submit his solution directly via the SSH tunnel to the *NeOS Backend Server* (Fig. 3, E) where it will be executed. This last phase can be repeated as often as the user wants to.

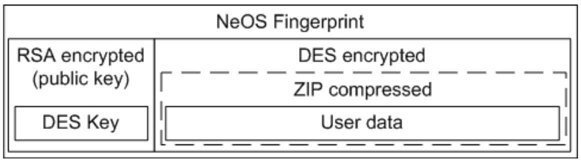

Figure 4. NeOS Fingerprint

2.2 Fingerprint

There are always some students which cannot resist to pass easily by just copying from their mates or cheating in some other way. These problems are even accentuated in an e-learning system, because of the lack of face-to-face interaction between students and tutors, the relative anonymity and the locality independence.
In the *VITELS* course, the plain text output of a program or the program code itself has to be copied into the course portal. This makes it very easy to manipulate the result. In order to satisfy the requirement of authenticity while still preserving compatibility we had to devise a test such that the user still may copy his solution into the course portal. We thus included a RSA/DES encryption as shown in Figure 4 to ensure that the students cannot change their results before copying them to the portal. Additionally, we are adding the user name provided by the SWITCHaai to create a user specific unique fingerprint. Also all data needed to reproduce the student results are included to enable the tutor to check the computer aided evaluation described in the next section.
The information in the Fingerprint consists of:

User name. Making the fingerprint unique for each user.

User modified files and random input. Enabling the tutor to reproduce the results.

Simulation output. Enables the tutor to check the computer aided evaluation.

Computer aided evaluation grade. Results of the computer aided evaluation.

For security reasons the public key has to be stored on the lab machine and the private key is only distributed to the tutors of the course. Also the fingerprints in the course portal can only be downloaded by the corresponding tutor.

2.3 Computer Aided Evaluation

It is very complex to fully check the correctness of a program. We thus restrict the control to verify partial correctness, only. Partial correctness is defined here as follows: the output of the execution of a program for a certain input must be correct. A reference implementation producing correct output for the test inputs is used to compare with the output obtained from the student's solutions. According to Hoare [8] a task is correctly solved, if the execution of both programs with the same input produces equal outputs. This simple method for evaluating the correctness of the solutions submitted by the students can be formally described as follows:

$$e(S,T) = \begin{cases} 1 & : \quad Q_A = Q_B \\ 0 & : \quad \text{otherwise} \end{cases} \tag{1}$$

with

$\{P\}S\{Q_A\}$

$\{P\}T\{Q_B\}$

P – Input

S – Reference Program

T – Student Program

Q_A, Q_B – Output

The output of both programs has to be significant, which means that that one must avoid that the check will only be successful for one particular input. Therefore, the input for the program has to be randomly chosen out of a certain interval. For instance, in our simulator for process scheduling the number of threads can vary between 3 and 10. The user can thus not manipulate his program to create a static output that would match with the output of the reference program, thereby falsely resulting in a positive evaluation.
Since Hoare's logic allows us, with the Rule of composition (Equation 2), to cut a program into several sequential pieces, we are also able to create a fine grained evaluation (Equation 3).

$$\frac{\{P\}R_1\{Q_1\}, \{Q_1\}R_2\{Q_2\}, \cdots, \{Q_{n-1}\}R_n\{Q\}}{\{P\}R_1; R_2; \cdots; R_n\{Q\}} \tag{2}$$

$$E(S,T) = \frac{\sum_{i=1}^{n} e_i(S_i,T_i)}{n} \tag{3}$$

Such a piecewise evaluation of sequential parts of a program renders the evaluation even more accurate.

2.4 NeOS Applet

The *NeOS Frontend Applet* in the *NeOS* is the principle user interface. The screenshot in Figure 5 shows an example with a common use case. In order to satisfy the requirements of ease of use as well as of flexibility (i.e. the tool must be able to handle multiple diverse use cases), we decided to create a multi layer tabbing area since allowing for every part to be accessible with less than three clicks.
The first tab layer (Fig. 5, A) consists of five tabs. The first one, the source code tab is the area where the user may access a text area (Fig.5, C). This area may be editable or not. For instance, the second tab layer in Figure 5 labeled B shows four tabs referencing different Java classes. Among those, only the first one is an editable Java class while the other three are just Java interfaces which may be read but not modified.
The second first layer tab is the console tab where all the output from the *NeOS Backend Server* will be presented. This includes all error messages (e. g. compiler error messages) not concerning the simulation or execution of the user source file. The grade the student would achieve for his program is also displayed in this window.
The third tab shows the user the random input with which his source code and the reference implementation were executed. This allows the user to assess the output of his program.
The next tab is the Simulator Output. As the name says, it shows the user the output of both simulations. Depending on the use case, it is also possible to show the user different output views, e. g. standard output or error output. These different views are then shown in the second layer tab. The second layer tabs will thus change as compared to the situation shown in Figure 5.
The last tab in the first layer is the pass code area. If the computer aided evaluation was successful, the fingerprint is displayed in the text area C. This can then be copied and pasted to the WebCT portal, thus completing the task.
The tab area (Fig. 5, D) at the bottom of the window contains the send button to start the submission and execution on the *NeOS Backend Server*, a cancel button to stop the execution, the help button for further information and a reset button to restore the original state again.

2.5 NeOS Server

The *NeOS Backend Server* is the heart of the *NeOS* system. It consists of core and task specific modules which are loaded at startup. Thus, for each task a new instance of the *NeOS Backend Server* needs to be started. We created the following interfaces to adapt the *NeOS Backend Server* to the actual task in an easy way:

Compiler interface: implements tasks like compiling the files

Simulator interface: implements the execution of user programs

Evaluation interface: implements the computer aided evaluation

Each interface contains methods which have to be implemented depending on the actual task. Furthermore, a configuration file has to be adapted to the actual needs. The configuration allows to adapt the applet, to set the path to the working directory and to define the *NeOS Backend Server* implementation which it has to use.

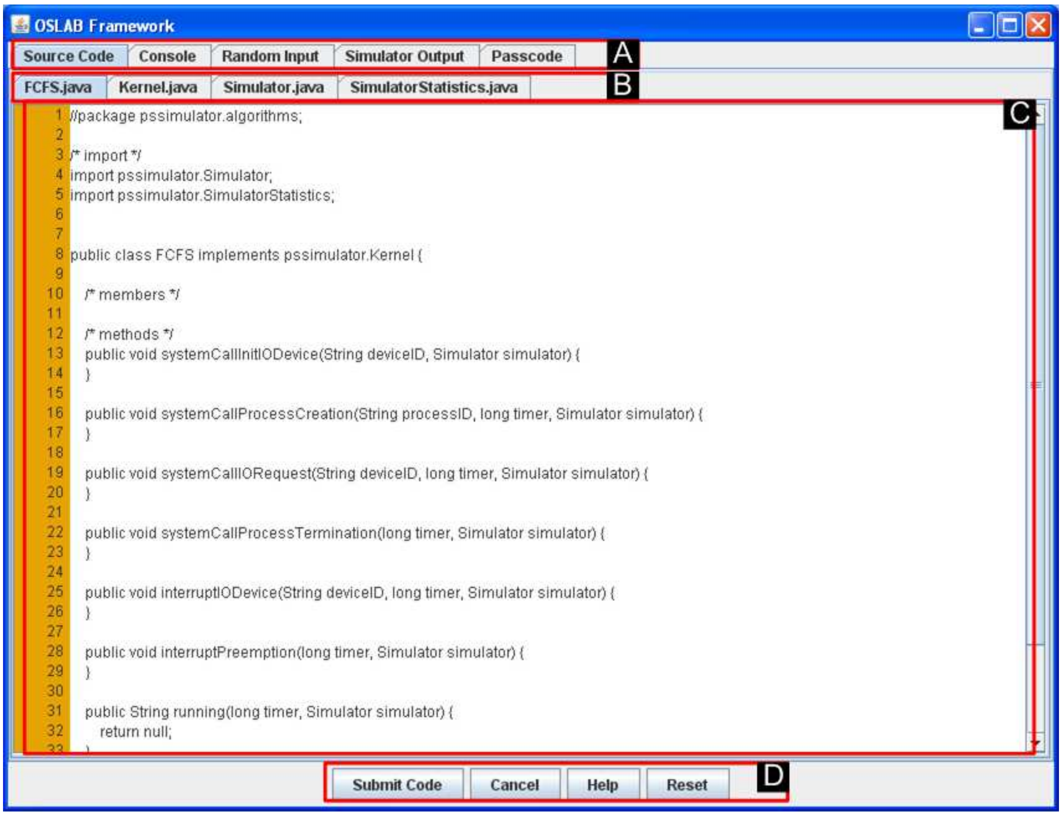

Figure 5. NeOS Frontend Applet

Since our *NeOS Backend Server* is written in Java all simulations should be started as an external process via the ProcessBuilder included in Java. This process starter provides the ability to start any other program or script written in any programming language. If the simulation software is written in Java, it is also possible to use the Java internal security manager to restrict the accessibility of the simulation. This includes access to the hard disk or access to the network connectors. If non-Java programs are used, the executed program has to be put in an external sandbox.

Since a program which compiles correctly does not necessary terminate upon execution, there was the need of an internal control mechanism. Therefore, we implemented as well a possibility to stop the simulation process after a certain time. So infinite loops will not disturb the system, and the program will just be terminated after a certain time depending on the use case.

Currently, the compiler interface supports the Java compiler. The other two interfaces concerning the simulator and the task evaluation depend on the particular tasks. A number of example implementations as well as some templates are available so far for these two interfaces. An implementation of the evaluation interface typically requires a method to compare the user program and the reference program outputs as described in Section 2.3. As an example, if eight out of ten output values are correct, a grade of 5 on a scale of 1 to 6 might be assigned (see Figure 6).

2.6 NeOS Tutor Tool

The *NeOS Tutor Tool* is intended to assist the tutor in evaluating the student's solutions. For every student, the tutor can review all the program code submitted as well as the input and output data, which are available through the students fingerprint as described in Section 2.2. Furthermore, the tool presents the results of the automatic evaluation. Figure 6 shows an example of the *NeOS Tutor Tool* user interface.

The results of the computer aided evaluation for a task are presented in a table (A in Figure 6). The tutor may now proceed to the final evaluation (tutor grade) by considering the grades provided automatically, by inspecting the program code submitted and the input respectively the output data, and by checking similarities between the different solutions of all the students.

For evaluating a student's solution, the tutor has to double click on the corresponding UserID, which opens up the user window (Fig. 6, B). There he has the possibility to grade the student and confirm it through the save button (Fig. 6, C). In the user window, all the details stored in the fingerprint can be reviewed through the tabs: the source code of the user, test input used for the simulation, and the simulation outputs (Fig. 6, D).

The different colors (or gray scales) in the UserID row

(Fig. 6, E) are a hint for the tutor that there are some parts in the solution of that student which are equal to the solutions of other ones. At the moment there are two colors (gray scales), red (dark) for a complete match and yellow (light) for a partial match. The checking is done file by file against all other submitted student solutions and the reference files. More details about this comparison can be obtained through the Similarities tab available in the user window (Fig. 6, D). An example of such a detailed view for the UserID ”Sabina” is given in Figure 7. This view can be easily adapted to the tutor's needs, since it is just an implementation of an interface as in the case of the ones of the *NeOS Backend Server*. At the moment, we just implemented a simple comparison which cuts out all uninteresting parts of the source code like spaces, tabs or braces. In our simple version the view shows for each file the UserID of the matches in a tree.

3 Conclusion and future work

In this paper, we have presented the *NeOS Framework*, which is an e-Learning tool developed to support especially hands-on training experiences. It consists of an Applet, a Server and a Tutor Tool, all implemented in Java. These tools allow for a user to locally program in an editor-like user interface and then test his source code on a well secured remote machine. At the moment the *NeOS Framework* is configured to work in the Swiss SWITCHaii and WebCT environment. It could, however, be adapted to any other environment by adding a corresponding user authentication mechanism.

An advantage of the *NeOS Framework* is the computer aided evaluation for programming tasks as well as the cross check of the student's solutions for detecting plagiates. Both features are meant to assist the tutor in his task to evaluate the student. Furthermore, it offers several security measures at the system level by securing the lab machines from being affected by attacks from the outside, and at the usage level by reducing the cheating possibilities.

Based on further experiences in using the *OSLab* course, we will evaluate the system for improvements. Already in development is a single server solution, such that one instance of the *NeOS Backend Server* can handle multiple tasks as well as a more sophisticated duplicate checking system. The present framework is restricted to tasks related to programming. We therefor plan for the future to generalize the framework for use in courses of other disciplines.

References

[1] Swiss Virtual Campus. http://www.virtualcampus.ch/, 2007.

[2] OSLab. http://www.oslab.ch/, 2007.

[3] Blackboard. http://www.blackboard.com/, 2007.

[4] SWITCHaai. http://www.switch.ch/aai/, 2007.

[5] VITELS. http://www.vitels.ch/, 2007.

[6] Apache Tomcat. http://tomcat.apache.org/, 2007.

[7] Shibboleth. http://shibboleth.internet2.edu/, 2007.

[8] C. A. R. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10):576–580, 1969.

[9] Java. http://java.com/, 2007.

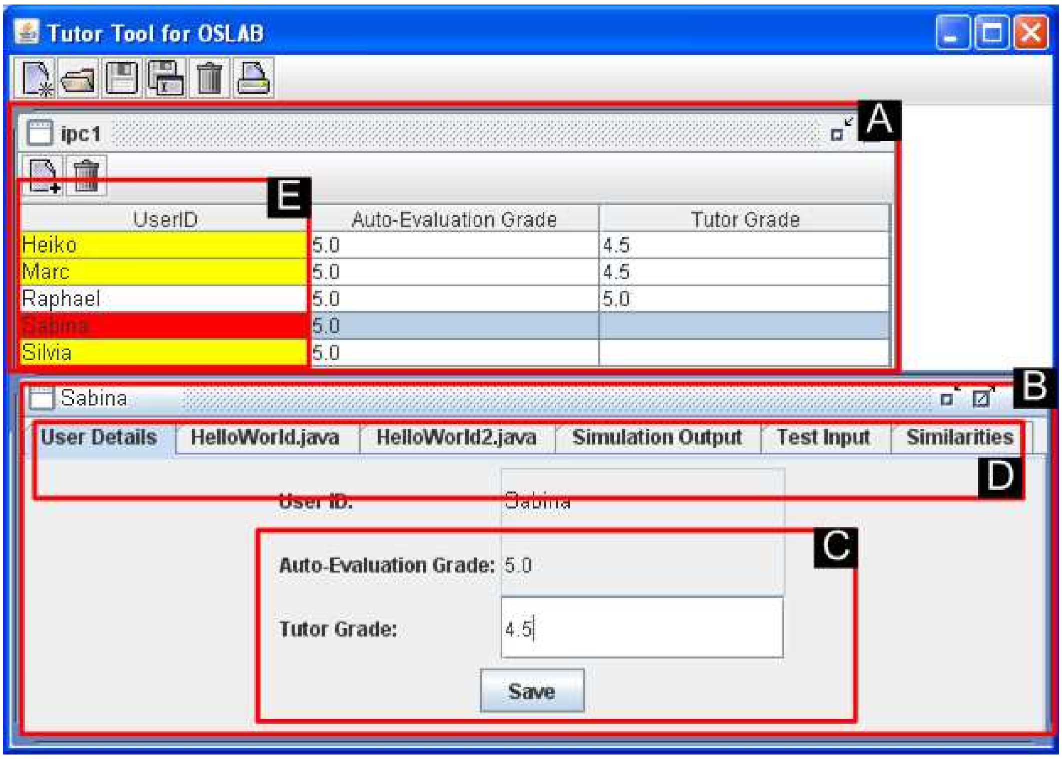

Figure 6. NeOS Tutor Tool

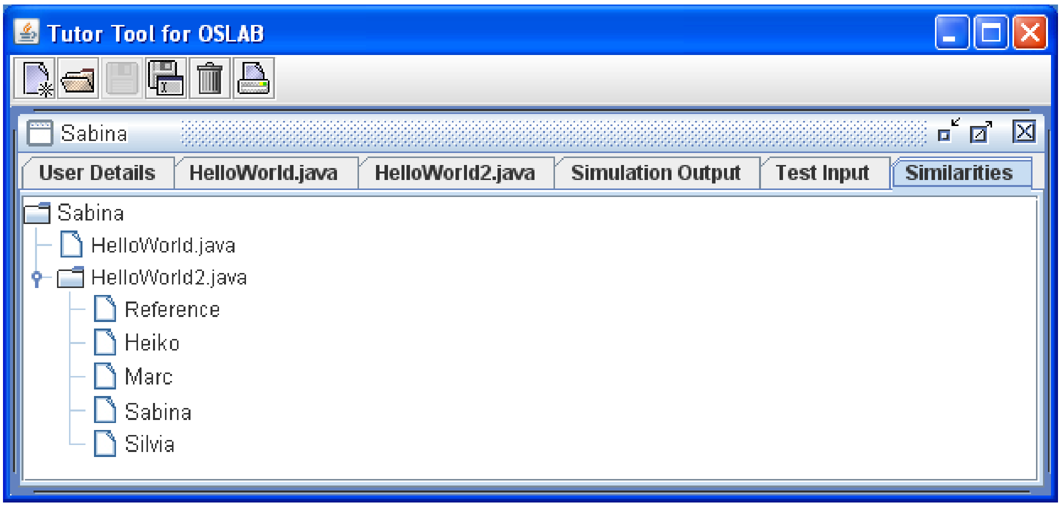

Figure 7. Similarity Tab of the NeOS Tutor Tool

A Flexible Instructional Electronics Laboratory with Local and Remote Lab Workbenches in a Grid

Ingvar Gustavsson
Blekinge Institute of Technology
Department of Signal Processing
SE-372 25 Ronneby, Sweden
ingvar.gustavsson@bth.se

Johan Zackrisson
Blekinge Institute of Technology
Department of Signal Processing
SE-372 25 Ronneby, Sweden
johan.zackrisson@bth.se

Kristian Nilsson
Blekinge Institute of Technology
Department of Signal Processing
SE-372 25 Ronneby, Sweden
kristian.nilsson@bth.se

Javier Garcia-Zubia
University of Deusto
Faculty of Engineering
Apdo. 1, 48080 Bilbao, Spain
zubia@eside.deusto.es

Lars Håkansson
Blekinge Institute of Technology
Department of Signal Processing
SE-372 25 Ronneby, Sweden
lars.hakansson@bth.se

Ingvar Claesson
Blekinge Institute of Technology
Department of Signal Processing
SE-372 25 Ronneby, Sweden
ingvar.claesson@bth.se

Thomas Lagö
Acticut AB
Gjuterivägen 7
SE-311 32 Falkenberg, Sweden
thomas.lago@acticut.com

Abstract

The Signal Processing Department (ASB) at Blekinge Institute of Technology (BTH) has created two online lab workbenches, one for electrical experiments and one for mechanical vibration experiments, mimicking and supplementing workbenches in traditional laboratories. Since some years, the workbenches are used concurrently with on-site ones in regular supervised lab sessions. The students are also free to use them on their own around the clock e.g. for preparation. The electronic workbench can be used simultaneously by many students. The aim of a project known as VISIR (Virtual Systems in Reality) founded by ASB at the end of 2006, is disseminating the online lab workbenches using open source technologies. The goal is to create a template for a grid laboratory where the nodes are workbenches for electrical experiments, located at different universities. This paper focuses on standards, pedagogical aspects, and measurement procedure requirements.

1 Introduction

During centuries scientists have performed physical experiments in order to verify and test theories and create proper mathematical models, describing reality well enough. Such experiments are the only way to “communicate” with nature and to learn its principles, often quite revealing. Only recently, it has been evident that mankind must live in symbiosis with nature and focus on sustainability and understanding. Thus, the demand for experimenters will increase. However, during recent decades the amount of hands-on laboratory work, for example, in engineering education has been reduced. The prime cause is clearly the task of handling the greatly increased student numbers, while staff and funding resources have scarcely changed [1].

Reducing the number of lab sessions is easy because laboratory work is seldom evaluated and the cost reduction obtained is often considerable. However, for example, ABET (Accreditation Board for Engineering and Technology) in USA has pointed out that learning objectives for laboratory work must exist and be evaluated [2, 3]. Thus, the amount of hands-on laboratory work in a course must be correlated to the learning objectives of the course. Unfortunately, a substantial rise of base funding resources is not probable.

It is, of course, also fundamental for students to understand theories and mathematical models. Often appropriate and low cost tools are hand calculations and simulations. The use of computer simulations has increased very much in engineering education in the last decades. However, to properly assess differences between mathematical models and real world, experiments are indispensable [4, 5]. On the other hand traditional laboratories have limited accessibility and high running costs.

Nowadays, students want an extended accessibility to learning resources and increased freedom to organize their learning activities which is also one of the main objectives of the Bologna Process. From a technological perspective, such flexible education corresponds to an adequate exploitation of information, communication devices and infrastructures, especially the Internet. Today, many academic institutions offer a variety of web-based experimentation environments so called remote laboratories that support remotely operated physical experiments [6-9]. This is one way to compensate for the reduction of lab sessions with face-to-face supervision.

The remote or online laboratories around the world are used in a variety of disciplines. However, the wide range of user interfaces is a problem for students and teachers. Efforts are being made to handle the situation. The iLabs project at Massachusetts Institute of Technology in USA, for example, has developed a suite of software tools that facilitates online complex laboratory experiments, and provides the infrastructure for user management [10]. A somewhat different approach would be to create a grid laboratory where the nodes are online lab workbenches distributed among a number of universities or other organizations. In such a laboratory intended for the same type of experiments it would be possible to organize supervised lab sessions with as many students or student teams working concurrently as are optimal for one instructor. Such supervised lab sessions could, for example, take place in a traditional laboratory where some students could use the local lab workbenches and others could perform the experiments remotely on distant grid nodes. Then it should be possible for each university to offer more time in the laboratory for its students.

In 1999, ASB started a remote laboratory project. Today ASB has two online lab workbenches, one for electrical experiments and one for mechanical vibration experiments, based on the BTH Open Laboratory concept [11]. The concept is about providing new possibilities for students to do laboratory work and become experimenters by adding online lab workbenches to traditional instructional laboratories to make them more accessible for students, whether they are on campus or mainly off campus. These workbenches are equipped with a unique interface enabling students to recognize on their own computer screen the instruments and other equipment most of them have previously used in the local laboratory.

At the end of 2006, ASB started the VISIR project together with National Instruments in USA and Axiom EduTech in Sweden, to disseminate the online laboratories at BTH using open source technologies. Axiom EduTech is a supplier of education, technical software, and engineering services for noise and vibration analysis. The project is financially supported by BTH and by VINNOVA (Swedish Governmental Agency for Innovation Systems).

What type of instructional laboratory would be feasible for creating a template for a grid laboratory? There are reasons for starting with a grid laboratory for electrical experiments:

- There are instructional electronics laboratories at most universities around the world containing the same equipment, (oscilloscopes, waveform generators, multi-meters, power supplies, and solderless breadboards) although models and manufacturers may vary. Such laboratories are already in a way a de facto standard.
- There are standards defining the functionality for instruments common in an electronics laboratory. The IVI Foundation is a group of end user companies, system integrators, and instrument vendors, working together defining standard instrument programming interfaces [12].
- Today BTH has an online electronics laboratory running in regular education where the software produced is released as open source [13].

Then this template can be used for designing grid laboratories for other areas. ASB has identified a laboratory for mechanical vibration experiments as an appropriate candidate because those lab workbenches are very expensive and the mathematical models are not accurate enough even for introductory courses.

2 The Open Electronics Lab at BTH

An experiment is a set of actions and observations, performed in the context of solving a particular problem. Experiments are cornerstones in the empirical approach to acquire a deeper knowledge of the physical world but also an important approach to verify that a model is accurate enough. The experimenter sets up and operates the experiment with his or her hands and/or with actuators. As an example, a lab workbench in an instructional laboratory for low-frequency analog electronics at BTH is shown in Fig. 1. The student wires a test circuit on the breadboard with the fingers and uses instruments to measure what s/he cannot perceive directly with the human senses as, for example, the electrical current. The experiments possible to perform in this environment are mainly limited by the set of components provided by the instructor.

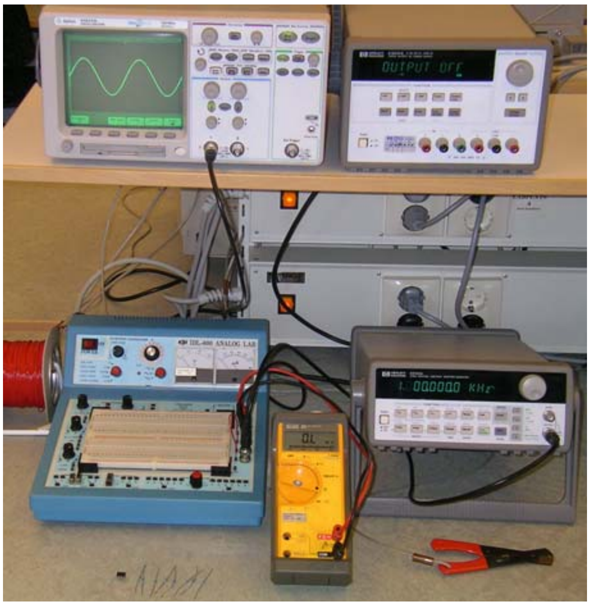

Figure 1: Workbench in a local electronics laboratory at BTH

In instructional laboratories at most universities, there are a number of lab workbenches where the same number of students or usually pair of students performs experiments supervised by an instructor. The students are permitted to be in the laboratories only during lab sessions when an instructor is present. The number of lab workbenches in a laboratory is usually selected, considering how many students an instructor can supervise if a workbench is not too expensive. Typically, electronics instructional laboratories are equipped with eight identical workbenches. Fewer lab workbenches mean more teaching hours per course but less investment. It is a pedagogical advantage if the lab workbenches are identical because the students can then perform the same experiments in each session and in the correct order required by the syllabus. On the other hand, it implies larger investments i.e. more duplicates of each instrument [9].

In electronics, it is possible to perform the same experiment in different time scales by selecting the electrical size of the components. This “feature” is used in the online electronics laboratory at BTH containing only one workbench to allow simultaneous access by time sharing. A single workbench can replace a whole laboratory with many workbenches. The maximum duration of a single experiment i.e. circuit creation and measurement procedure is currently set to 0.1 second to get a reasonable response time even with a large number of experimenters. The experiments are set up locally in each client computer. Only by pressing a *Perform Experiment* button the experimenter sends a message containing a description of the desired circuit and the instrument settings to the workbench (server). If the workbench is not occupied, the experiment procedure is performed in a predefined order, and the result or an error message is returned to the requesting client computer. Otherwise, the request is queued.

The online lab workbench at BTH is different versus the traditional one in Fig.1. It is, of course, not possible for students to manipulate the components and wire a desired circuit on the breadboard with their fingers remotely. A type of circuit-wiring robot e.g. a relay switching matrix must be used. The instruments are plug-in boards installed in the PXI chassis connected to the host computer as shown in Fig. 2. This chassis and its contents are manufactured by National Instruments. The corresponding virtual front panels are photographs of the front panels of the instruments in Fig. 1. As an example, a screen dump displaying the multi-meter is shown in Fig. 3. The card stack on the top of the PXI chassis in Fig. 2 is the switching matrix. A subset of the components a teacher or the laboratory staff has installed in the matrix is displayed on the client computer screen adjacent to a virtual breadboard where the student wires the desired circuit to control the matrix. It is possible to assemble a circuit with up to 16 nodes by engaging a number of relays in the matrix. Apart from a controller board the card stack contains two types of board: one with component sockets and one for connecting instruments. The nodes passing all boards can be connected to sources, instruments, and/or components installed in the sockets via relay switches. The online electronics laboratory at BTH is used in three ways:

- In supervised lab sessions in the local laboratory where the students can select if they want to perform the experiments locally or remotely. However, in the first lab session it is mandatory to do the wiring on the real breadboard.
- In supervised lab sessions in distance learning courses where the students are scattered all over the country. Various communication methods are used to communicate between the students themselves and between the students and the instructor.
- Students can prepare supervised lab sessions and perform the experiments at home knowing that the equipment in the traditional laboratory look the same and behave in the same way. They can also repeat experiments afterwards. Especially inexperienced or less confident students requiring more time appreciate these

possibilities. A student wanting, for example, to master the oscilloscope can practice at home without anybody watching.

Figure 2: Equipment Server

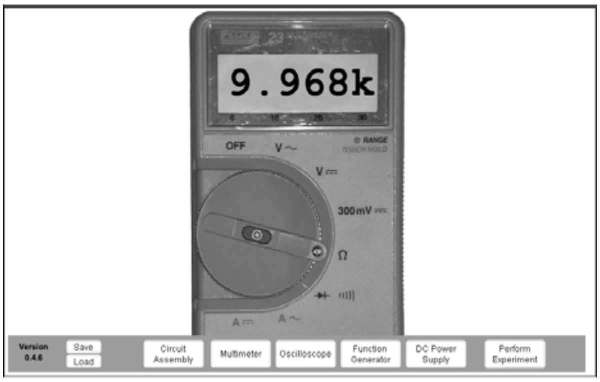

Figure 3: Screen dump showing the multimeter

So far the research has been focused on recreating as accurately as possible the laboratory experience for the remotely based learner. Remote desktop software and MS Messenger has been used for communication. More advanced communication means will be adopted [14].

3 The VISIR project

The aim of the VISIR project is to form a group of cooperating universities and other organizations, a VISIR Consortium, creating/modifying software modules for online laboratories using open source technologies and setting up online lab workbenches [15]. A number of such scattered lab workbenches may be nodes in a grid laboratory. The VISIR Initiative is not confined to electronics laboratories but the VISIR project has started with lab workbenches for electrical experiments, since that is an easy and straightforward application to demonstrate the powerful concept. So far, the following universities are participating or are interested in participating in the project FH Campus Wien in Austria, University of Deusto in Spain, University of Genoa in Italy, Princess Sumaya University for Technology in Jordan, Carinthia University of Applied Sciences in Austria, Gunadarma University in Indonesia, UNINOVA (Institute for the Development of New Technologies) in Portugal, and ISEP (Instituto Superior de Engenahria do Porto) in Portugal. The first two universities have already implemented online workbenches using the currently released software. BTH will act as a hub for the development and maintain a server from which the current version of the software can be downloaded.

The overall goal of the VISIR project is aimed at increasing the access to experimental equipment in many areas for students, without raising the running cost per student significantly for the universities. The means are shared online laboratories created by universities in cooperation and supported by instrument vendors. Sharing of laboratories may lead to sharing of course material. The ultimate goal of our research at BTH is ubiquitous physical experimental resources, accessible 24/7 for everyone, gender neutral, as a means of inspiring and encouraging children, young people and others to study engineering and become good professionals or to be used as a means of life-long learning.

4 A grid laboratory for electrical experiments

Grid computing has emerged as a way to harness and take advantage of computing resources across geographies and organizations. Grid architecture for an electronics laboratory similar to the BTH one has been published [16, 17]. In this grid-based laboratory a measurement workflows execution service takes care of executing the measures according to the rules and sequence described in a measurement workflow repository. It invokes instrument services and manages multi-user concurrent sessions on the same physical test bench. The composition of measurement workflows is in charge to teachers, who provide the description of the measurement process in terms of, for example, instruments activation process. Sharing one lab workbench is the same approach as in the current online laboratory at BTH. On the other hand knowing how to handle the measurement process is an important part of lab assignments. To display a transient on the oscilloscope, for example, the oscilloscope must first be armed and then the transient is started. Each student or student team in front of a client computer should have a workbench at their own disposal for exclusive access as in the local laboratory. Then the *Perform Experiment* button is no longer required.

It should be possible to organize a grid laboratory distributed among universities around the world. The workbenches should be the proper grid nodes. Smaller nodes are not feasible because the instruments and the circuit under test must be located close together. The instruments and the circuit creation manipulator would be device services accessible by the lab clients via virtual front panels or a virtual breadboard, Fig. 4, 5. Web services prescribe XML-based messages conveyed by Internet protocols such as SOAP. However, real time performance requires protocols without significant latencies and overhead. For example, the oscilloscope display should be updated at least every second.

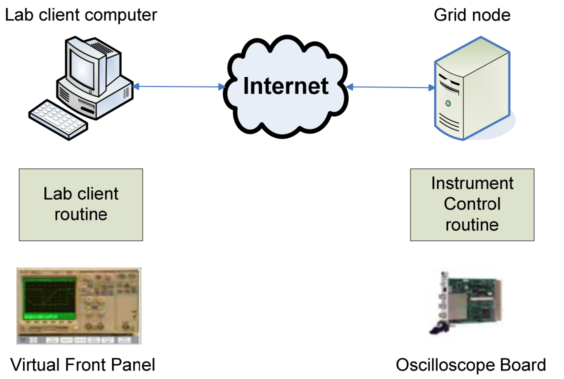

Figure 4: Oscilloscope service

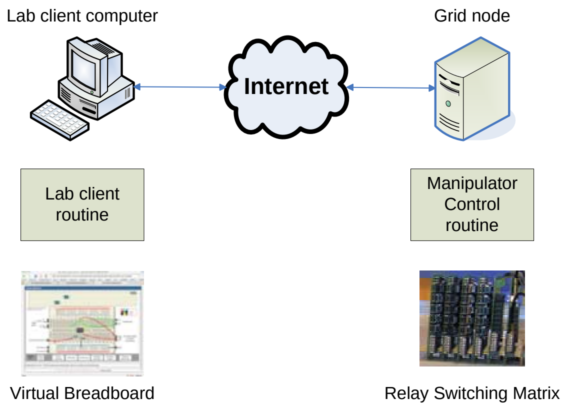

Figure 4: Wiring service

It is possible to combine a virtual front panel representing a particular instrument from one manufacturer with the corresponding hardware from another as long as the performance of the hardware matches that of the depicted instrument. The VISIR client software package is modular and it is recommended that every university creates virtual front panels representing the instruments they have in their local laboratories to preserve the student's context.

Instrument I/O is a well-studied domain with established industrial standards. Most commercial products follow the Virtual Instrument System Architecture (VISA) or the Interchangeable Virtual Instrument (IVI) standards [18]. The IVI foundation creates instrument class specifications. There are currently eight classes, defined as DC power supply, Digital multi-meter (DMM), Function generator, Oscilloscope, Power meter, RF signal generator, Spectrum analyzer, and Switch. Within each class a base capability group and multiple extension capability groups are defined. Base capabilities are the functions of an instrument class that are common to most of the instruments available in the class. For an oscilloscope, for example, this means edge triggering only. Other triggering methods are defined as extension capabilities. For example, the functions supported by the VISIR oscilloscope are listed in Table 1. The goal of the IVI Foundation is to support 95% of the instruments in a particular class.

Table 1: The VISIR oscilloscope capabilities

Group Name	Description
`IviScopeBase`	Base Capabilities of the IviScope specification. This group includes the capability to acquire waveforms using edge triggering.
`IviScopeWaveformMeas`	Extension: IviScope with the ability to calculate waveform measurements, such as rise time or frequency.
`IviScopeTriggerModifier`	Extension: IviScope with the ability to modify the behavior of the triggering subsystem in the absence of a expected trigger.
`IviScopeAutoSetup`	Extension: IviScope with the automatic configuration ability.

It is not necessary to use IVI drivers but to enable interchangeability between grid nodes VISIR recommends functions and attributes defined by the IVI Foundation to be used to describe the capabilities of the lab hardware. In this way it should be possible to create a standardized approach which is easy to adopt.

5 Conclusions and future work

BTH is disseminating software for an online workbench comprising the same equipment as a workbench in a traditional instructional electronics laboratory. Two universities have already implemented such workbenches using the VISIR software. A number of students can perform experiments on each of these online workbenches simultaneously by time sharing. However, each remote student or student team should have a workbench at their own disposal to be able to control each step of the measurement process. A way to reach this ideal situation would be increasing the number of online workbenches and organize them in a grid. Further research seems to be required to get real time performance comparable with that of the local workbench when a web service approach is to be adopted. The goal is to offer free access to experimental equipment for students and a lab experience that is as genuine as possible despite the lack of direct contact with the actual lab hardware. The universities will also be able to produce experimenters without increased running cost per student.

References

[1] D. Magin and S. Kanapathipillai, “Engineering Students’ Understanding of the Role of Experimentation”, *European Journal of Engineering Education*, 2000, Vol. 25, no. 4, pp. 351-358

[2] L. D. Feisel and A. J. Rosa, “The Role of the Laboratory in Undergraduate Engineering Education”, *Journal of Engineering Education*, January 2005, pp 121-130.

[3] Cooper, M., “Remote laboratories in teaching and learning – issues impinging on widespread adoption in science and engineering education”, *International Journal of Online Engineering*, Vol. 1 No. 1, 2005.

[4] Nedic, Z., Machotka, J., and Nafalski, A., “Remote Laboratories Versus Virtual and Real Laboratories”, *Proceedings of the 33rd ASEE/IEEE Frontiers in Education Conference*, Bolder, USA, November 5 – 8, 2003.

[5] J. Ma, and J. V. Nickerson, "Hands-on, simulated, and remote laboratories: A comparative literature review", *ACM Computing Surveys*, 2006.

[6] D. Gillet, A. V. N. Ngoc, and Y. Rekik, “Collaborative Web-Based Experimentation in Flexible Engineering Education”, *IEEE Transactions on Education*, Vol. 48, No. 4, November 2005 .

[7] Z. Nedic and J. Machotka, “Remote Laboratory NetLab for Effective Teaching of 1st Year Engineering Students”, *Proceedings of the REV 2007 Conference*, Porto, Portugal, June 25 – 27, 2007.

[8] A. M. Scapolla, A. Bagnasco, D. Ponta, and G. Parodi, “A Modular and Extensible Remote Electronic Laboratory”, *International Journal of Online Engineering*, Vol. 1 No. 1, 2005.

[9] J. Garcia-Zubia et al., ”WebLab-GPIB at the University of Deusto”, *Proceedings of the REV 2007 Conference*, Porto, Portugal, June 25 – 27, 2007.

[10] iLabs: Internet access to real labs - anywhere, anytime, http://icampus.mit.edu/iLabs/, 2007-07-20.

[11] L. Gomes and J. Garcia-Zubia (eds), Chapter 11 in *Advances on remote laboratories and e-learning experiences*, University of Deusto, Bilbao, Spain, 2007, pp. 247 – 267, ISBN 978-84-9830-077-2.

[12] http://www.ivifoundation.org/, 2007-12-15.

[13] I. Gustavsson 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.

[14] MJ. Callaghan, J. Harkin, TM. McGinnity and LP. Maguire, “Paradigms in Remote Experimentation”, *International Journal of Online Engineering*, Vol. 3 No. 4, 2007.

[15] I. Gustavsson et al., “The VISIR project – an Open Source Software Initiative for Distributed Online Laboratories”, *Proceedings of the REV 2007 Conference*, Porto, Portugal, June 25 – 27, 2007.

[16] A. Bagnasco, A. Poggi, A. M. Scapolla, “A Grid-based Architecture for the Composition and the Execution of Remote Interactive Measurements, ”*2nd IEEE International Conference on e-Science and Grid Computing*, Amsterdam, the Netherlands, Dec. 2006.

[17] A. Bagnasco, A. Poggi, A. M. Scapolla, “Computational GRIDSs and Online Laboratories”, *1st International ELeGI Conference on Advanced Technology for Enhanced Learning*, 2005.

[18] Y. Yan, Y. Liang, X. Du, H. Saliah-Hassane, and A. Ghorbani, ”Putting Labs Online with Web Services”, *IT Pro*, March/April 2006, Published by the IEEE Computer Society.

Simulation of an Intelligent Network - Basic Call State Model

Remote Laboratory

Florin SANDU
"Transilvania" University of Brasov
Faculty of Electrical Engineering and Computer Science
Str. Politehnicii nr. 1, Brasov, 500024
ROMANIA
iesc.unitbv.ro
sandu@unitbv.ro

Sorin BÂLBÂE
sorin.balbae@siemens.com

Szilárd CSEREY
szilard.cserey@siemens.com

Titus Constantin BĂLAN
titus.balan@siemens.com

Siemens Program and System Engineering
Str. Mihail Kogalniceanu nr.21, Brasov, 500090 ROMANIA
www.pse.siemens.ro

Abstract

This paper presents a remotely accessible Telecom Intelligent Network (IN) simulator, centered on the Basic Call State Mode (BCSM). This model is a Finite State Machine which describes the behavior of IN services in the call processing. The simulator is dedicated to the students in Telecom and allows them to get acquainted with the operation of the IN Service Control Point (SCP) in different scenarios they can imagine.

1 Introduction

Intelligent Networks were created to enhance and to diversify the telecommunication services in fixed and mobile networks.

Considering the architecture of "service switching" that is put "on-top" circuit switching and packet switching, IN is an add-on to a "service-independent" telecom network. That is, intelligence is taken out of the switches and placed (as "middleware") in separated computer nodes that are distributed throughout the network. This provides the network operator with the means to develop and control services more efficiently, with minimum dependencies of the infrastructure. New capabilities can be rapidly introduced into the network.

Once introduced, services can be easily customized to meet individual client's needs.

2 Service Creation, Service Creation Environment

Many Intelligent Network software vendors have completed service creation software with state-of-the art computer graphics, to eliminate the need for traditional programming methods. After some practical studies, IN software vendors found out that it is better to develop services in a graphical way because in this way it becomes not only more clear but also accessible to "service integrators" that can be more rather better acquainted to usability of services (according to end-customers' social and cultural profile) than to software development. Through the use of menu-driven software, services are created by introducing various service parameters.

Figure 1 below provides an example of a service-independent-building-block (SIB) approach to create IN services. Play announcements, collect digits, call routing, and number translation SIB-s are shown.

The SSP (Service Switching Point) is basically a SSS (Switching Sub-System) enhanced with extra digit processing (to detect the "IN codes" in the prefix of the dialed digits). The SSS capability to play announcements is also essential for the IN. Routing the call is an SSP function.

Number translation is an SCP capability.

By arranging these four capabilities (and these corresponding SIB-s) in various combinations, services such as 0800 with interactive dialing, outgoing call screening, and area number calling can be created.

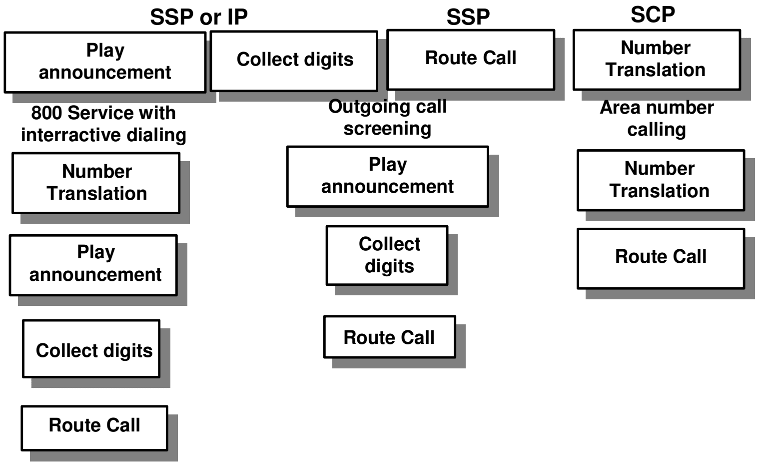

Figure 1. Service-independent-building-blocks (SIB) and their localization

The IN standard was defined by ITU (International Telecommunication Union) and proposed that every functionality of the IN should be separated into different elements, allowing this way the addition of other elements (SIB-s made by other vendors) to the IN, and the interoperation of IN elements in the same system. Services can be developed using different combinations of SIB-s, leveraging this way the services analysis and their interoperation, as well as pre-testing before their implementation.

SIB-s are described by the ITU-T standard as being abstract procedural blocks which can process a number of inputs to change the behavior of the service and to generate local data.

3 Using LabView as a "service creation tool" and its educational benefits

We have implemented a didactical Basic Call State Model using National Instruments LabView.

The main reasons we have used LabView are:

- It is a graphical development environment, being very close to commercial tools dedicated to service creation LabView has the excellent "merge" of logical diagram – the so-called software organigram – and of hardware block-schematics. In fact, it allows a pure co-design, and only after this the project can be implemented in hardware/firmware/netware/software according to technological availabilities.
- It includes VI Templates (.VIT files pre-fabricated) that are very well suited for finite-state-machines (the "Standard State Machine" VIT with the diagram of figure 2).
- It gives the possibility to extend the functionality of an existing service application by the interconnection of virtual instruments (VI-s); every VI, for example, could represent a SIB).

With this project our goal was to develop a simple example of IN Finite State Machine, that students can test in different own scenarios.

Our implementation of the Basic Call State Model supports the "Free-Phone" service, the number that is implemented in the simulator is 0800111.

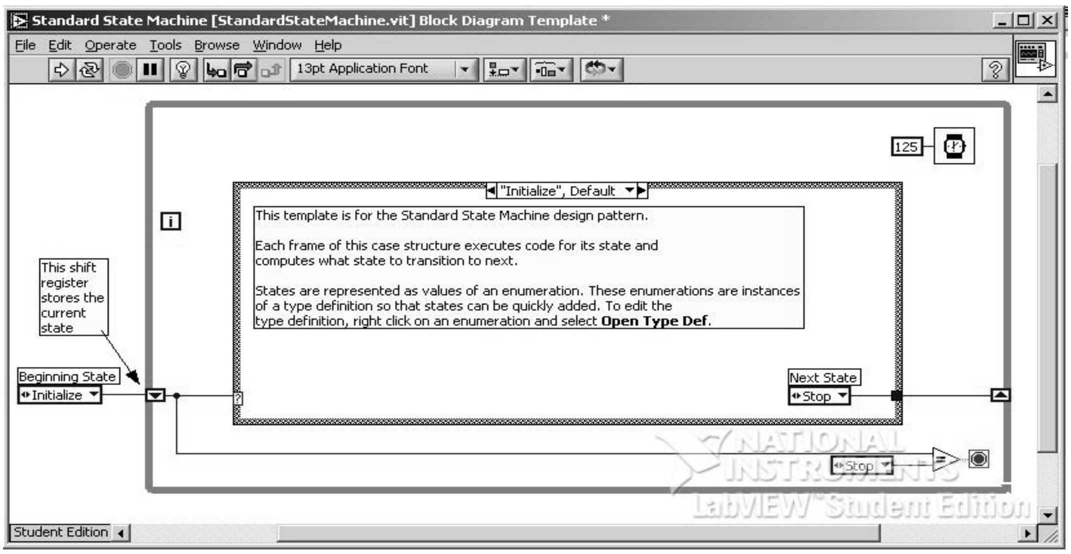

Figure 2. Diagram of the "Standard State Machine" VIT

4 Remote Simulation and laboratory, server deployment

The BCSM simulator resides in a Microsoft Windows 2003 server, and can be accessed via Internet at http://82.78.149.213:30700/identity1.html

LabView has an integrated web server which allows different kind of VI-s panels to be viewed and controlled remotely. In order to successfully deploy the web-server the following configurations are needed:

- set the port number of the web-server
- choose the “VI” file that contains the simulation you want to run
- complete the information text boxes of the “web publishing tool”: name of the simulation, copyright and so on.

One has to take into consideration that the LabView web server is set by default to use port number 80 (for the http protocol) but if another "main" web server (for example Microsoft IIS) that also uses port 80 has to run on that machine, then LabView web server has to run on another port.

5 Operation of the simulator

The application has two modes of user interaction:

- One is when the user is not implied into the handling of events, it is just an observer. To enter this mode one has to leave un-checked the “User interaction needed” checkbox (see figure 3).
- The other mode, more interactive, needs the user to select some of the decisions in order to handle a call. These decisions are called Detection Points (DP-s).

PIC-s (Points In-Call) represent the call processing state of the flexible service logic that emulates the SCP, i.e. a running procedure in the basic call process.

Figure 3. User interaction option in the BCSM.VI panel

DP-s (Detection Points) are points between the PIC-s at which the call processing may be suspended in order to invoke actions of the service logic (they are associated with the interrupt events that are taking the BCSM to- our out- of a state) i.e. asking for new instructions or data ("parameters") before the call processing continues or sending notifications to the flexible service logic.

Figure 4 presents the two (interconditioned) BCSM: the Originating BCSM (O-BCSM) that represents the behavior of the IN at the calling party side and the Terminating BCSM (T-BCSM) that operates at the called party side (that can be very far away from O-BCSM; nevertheless, some signaling is needed between O- and T-BCSM).

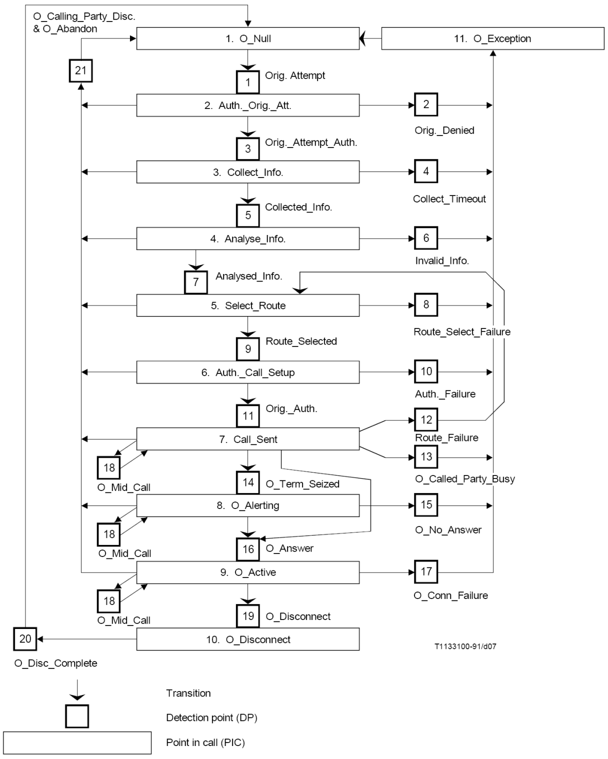

Figure 4.a. The Originating BCSM flow diagram

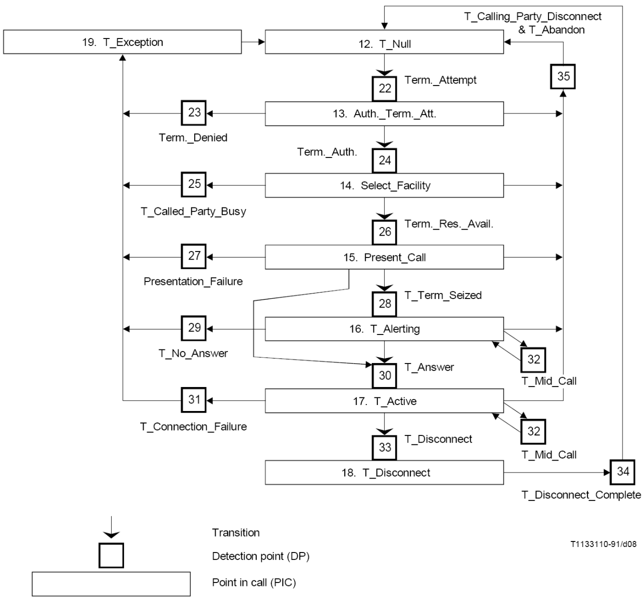

Figure 4.b. The Terminating BCSM flow diagram

The BCSM "demonstrator" was done by implementing one representative service (the “Freephone”) and the toll free number that has to be called is 0800111

If the destination number ("Called Party" or "B-Party" number dialed has the prefix of 0800 then the BCSM will analyze :

- The day of the week
- The time interval (inside three shifts of the day)
- The digits of the suffix (the numbers coming after the prefix)

According to this analyzed information it will take the corresponding decisions on routing the call to the physical numbers available (and implemented in the "routing table" that is the database associated to the "Freephone" service in the SCP).

As the digit analysis includes also the processing of the "Calling Party" (or "A-Party") number, the IN will route the call according to the following rules:

- "Origin-dependent" routing: the call is routed to the telecom switch that is closest to A-Party (they were considered, as an example, three Romanian county capital towns, Brasov, Braila and Timisoara, as well as the capital of Romania, Bucharest and two main telecom operators, RomTelecom and RDS-RCS (see figure 5)
- "Time-dependent" routing:
  - If it is Sunday the routing will be made automatically to the capital of the country and the called number 0800111 will be translated to 021080080 or to 031080080 if A-Party had a RomTelecom or RDS-RCS number respectively (than origin dependency was also considered).
  - If it is Saturday, the routing will be made "origin-dependent" (considering the A-Party number's prefix, <area code>) to the closest regional telecom switch, but the called number 0800111 will be translated only to the number <area code>0800800
  - If it is one of the working days (Monday to Friday), the routing is done according to the A-Party - caller's - number prefix and to the time interval (time shift). The called number 0800111 will be translated to <area code>080080<the number representing the time shift - 1, 2 ,or 3>

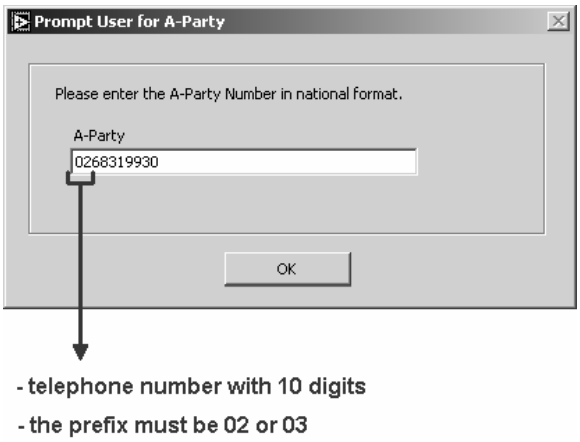

Figure 5. Example of an A-Party number

6 Examples of how students can test the calls

In order to make a traditional call (to RomTelecom or RDS-RCS numbers), without using specific IN numbers (with prefixes – "IN codes" that represent different IN services), one has to use the prefix of “02” or “03” (respectively) – which are area codes – and 8 additional digits in order to have a correct and complete telephone number. So, a normal telephone number has to have 10 digits and the traditional routing of the call will be "destination-dependent". If the prefix of the B-Party (called number) has one of these two values (02 or 03) then the call is considered to be a simple one, that doesn’t use IN services. An example of a regular call (without the use of an IN service) is given in figure 6:

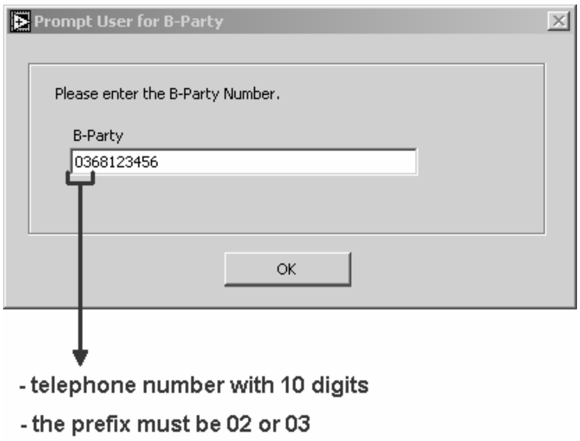

Figure 6. Case of B-Party called by normal dial-in

If the prefix of the called number is different from 02 or 03, then the BCSM will check if it is 0800 or something else. To invoke the BCSM, the prefix has to be 0800 because this is the only service that is implemented in the simulator, namely it is the “Freephone service” and the telephone number must additionally contain between 3 and 6 digits after the 0800 prefix.

The "Freephone" number (toll-free number) must have the prefix 0800 and from 3 to 6 digits. The only number that is implemented is 0800111 (see figure 7.a), so dialing a similar but different number will result in the message of figure 7.b.

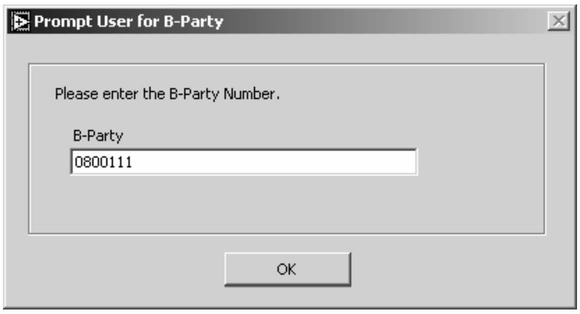

Figure 7.a. B-Party dial-in as IN number

Figure 7.b. Message in case the student dialed B-Party as (similar but) wrong IN number

The panel of the BCSM.VI is given in figure 8.

Auto-correction of numbers (by truncation of last digits or, on the opposite, adding zeros in place of un-dialed last digits) was also implemented. If this option isn't chosen, the student will be invited to re-dial wrong numbers (see figure 9).

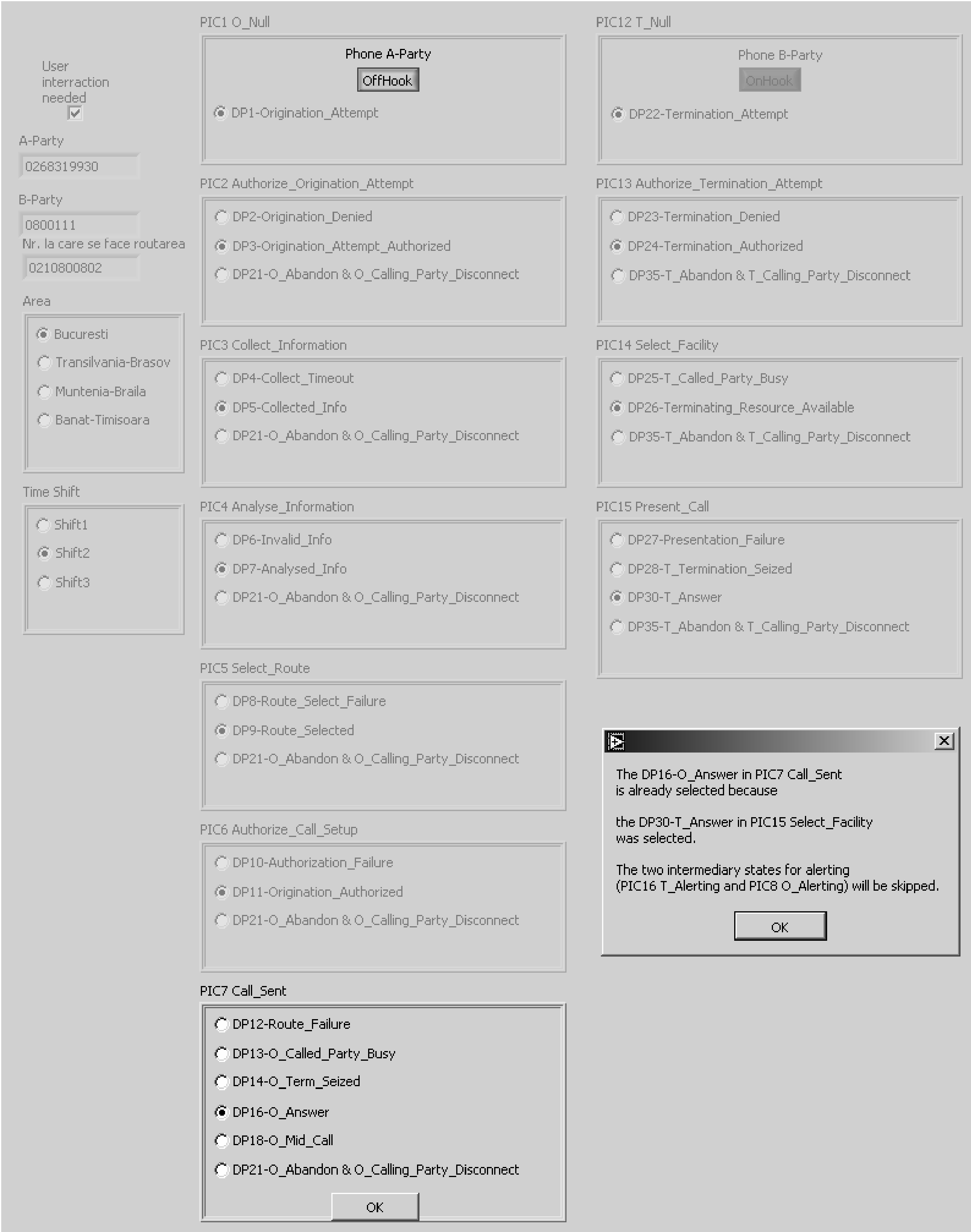

Figure 8: Panel of the BCSM.VI

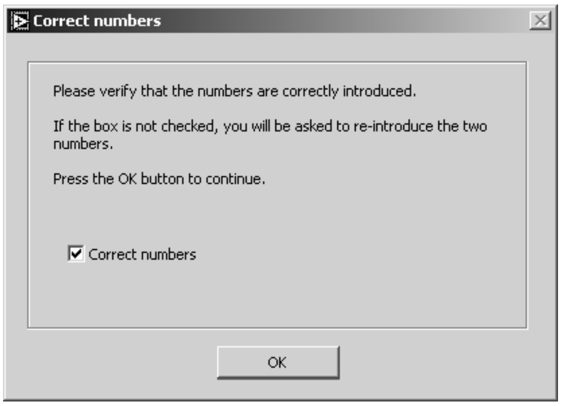

Figure 9. Message for the auto-correction of numbers

7. Conclusions and further development

The authors synthesized didactical experience, accumulated at "Transilvania" University of Brasov – Romania, with telecom experience offered by Siemens Program and System Engineering Romania which is a real "Teaching Company".

LabView was preferred as the most outstanding "instrumental IT" development and test environment. Its good visualization enables students to see, in step-by-step mode, animated flow of information through a system, with instant values that are plotted in the nodes, giving the possibility to pause and analyze the conditioning.

Remote access is very easy via Internet as LabView "Run-Time Engine" is automatically plugged-in most of the browsers. The authors programmed the automatic Grant Control at the LabView web server and let the BCSM.VI to run continuously.

IN behavioral models can be further extended with new SIB-s in the university laboratory works, and further on, new services can be implemented.

References

[1] H. Dobernig – Siemens Program and Systems Engineering - Vienna, Austria – *"Intelligent Packet Switching for GPRS / UMTS Core Switches"* – Proceedings of the Workshop on "Wearable and Pervasive Computing" – International Conference on "Telecommunications and Mobile Computing" – TCMC 2005, Graz, Austria.

[2] F. Sandu, D.M. Istrate - *Unified Benchmarks for Mobile Intelligent Networks* – Proceedings of the 36th International Symposium of the Romanian Military Equipment and Technologies Research Agency – "METRA" - 26-27 May 2005, Bucharest, ROMANIA, ISBN 973-0-03923-2

[3] G.S. Blair, G. Coulson – *"Service Architectures"*, Chapter in *"Communications Services – The Technology of Call Control",* ed. Kenneth J. Turner, Evan H. Magill and David J. Marples, John Wiley and Sons, ISBN 0-470-85066-3, 2004

[4] I. Venieris, H. Hußmann (eds.) – *"Intelligent Broadband Networks"* – John Wiley, 1998.

[5] Wilkinson,N.: *Next Generation Network Services. Technologies and Strategies*. John Wiley and Sons, 2002

SOA Meets Robots - A Service-Based Software Infrastructure For Remote Laboratories

Peter Tröger
Blekinge Institute of Technology
Ronneby, Sweden
peter.troger@bth.se

Andreas Rasche
Hasso-Plattner-Institute
at University of Potsdam, Germany
andreas.rasche@hpi.uni-potsdam.de

Frank Feinbube
Hasso-Plattner-Institute
at University of Potsdam, Germany
frank.feinbube@hpi.uni-potsdam.de

Robert Wierschke
Hasso-Plattner-Institute
at University of Potsdam, Germany
robert.wierschke@hpi.uni-potsdam.de

Abstract

With the ongoing internationalization of virtual laboratories, the integration of such infrastructures becomes more important. The meanwhile commonly accepted 'glue' for such legacy systems are service-oriented architectures, based on standardized and accepted Web service standards.

We present our concept of the 'experiment as a service', where the idea of service-based architectures is applied to virtual remote laboratories. In our laboratory middleware, experiments are represented as stateful service implementations. We discuss performance, reliability and security in this approach, and show how our solution - the Distributed Control Lab - is applied in the European VetTrend project.

1 Introduction

The Distributed Control Lab (DCL) is a virtual laboratory at the Hasso-Plattner-Institute in Potsdam, which enables the remote usage of experiments for teaching purposes. Authenticated users can submit control programs for an experiment by the help of different front-ends, such as the Web interface, a development environment plugin or a command-line interface. Each control program by a particular user is called a *job*, which is executed by a matching *experiment controller* that steers the according physical experiment hardware.

The DCL infrastructure is responsible of distributing incoming jobs to available experiments. Multiple experiments of the same type, being able to handle the same kind of job, are called *experiment types*. The infrastructure supports both physical experiments and simulations for the same experiment type. Simulations are intended to help out in case of high load on experiments, e.g. before a student assignment deadline. Simulations can act as full replacement for the real experiment in most cases, since students submit most of their jobs for checking the correctness of their control application. This only demands mainly a compiler run for the control program in the particular runtime environment, but not a real execution of physical activities [4, 6].

Beside the support for teaching activities, research in the DCL project covers the question of protecting the infrastructure against malicious code, which can potentially harm experiment hardware or execution nodes. It utilizes techniques such as automated source code analysis, run-time monitoring and dynamic adaptation for protecting the experiment infrastructure [5].

Within the DCL, several real-time control experiments have already been integrated. Figure 1 gives an overview of some experiments connected to the DCL. Foulcault's Pendulum as an experiment imitates Leon Foucault's famous experiment for measuring the earth rotation. An iron ball is used for the pendulum that can be accelerated using an electro-magnet connection to a control-PC via USB. Two orthogonal laser-based light barriers provide position information about the swinging ball. In this experiment, students have to implement an algorithm which evaluates the light barriers and switches the magnet on and off to keep the pendulum swinging.

A second experiment is the Higher Striker, which works like a linear motor. Seven electro-magnets are placed around a tube of glass and can be used to accelerate an iron cylinder. Light barriers among the tube can be used to determine the position of the cylinder. The task of the experiment is to analyze the data-stream sampled from the light-barriers and generate a control data-stream for the magnets to move the cylinder to the top of the tube. The electro-magnets and the light barriers are sampled by a control-PC with a fre-

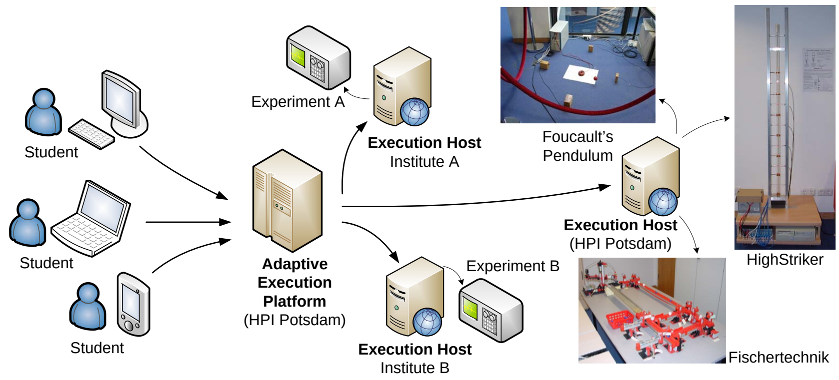

Figure 1. The Distributed Control Lab

quency of 38,4 KHz. We use this experiment to teach the programming of embedded real-time control applications and students compare different real-time operating systems on the control-PC.
Another experiment shown in Figure 1 is a model of an assembly line controlled by a programmable logic controller (PLC). In addition to the control program, which has to be implemented in a PLC-language (IEC 61131), monitoring and HMI components are implemented as Java and .NET programs, which be uploaded via our laboratory infrastructure. Students can use this experiment to experience heterogeneous embedded control systems. Further experiments also include the programming of Lego NXT robots and various simulators for our physical experiment installations.

1.1 Motivation

For several years, the DCL installation at HPI was based on a proprietary distributed .NET application. Both the experiment controllers and the job scheduler where realized with the .NET 1.1 framework, where components interact through a proprietary remoting technology. Meanwhile, new intended experiment types require the experiment controllers to be implemented in Java and other languages that are either not or badly supported by the .NET environment. This requirement motivated the usage a service-oriented middleware, in order to integrate different execution platforms for the experiment controllers.
Within the Vet-Trend project, we are also investigating the integration of experiment installations from different organizations. Since different technologies and execution platforms are in use in the field, a way must be found to integrate these heterogeneous systems. This requirement motivated the usage of commonly accepted Web Services standards to access the experiment installations.
Another motivation to improve the architecture of the DCL was the separation of compile and execution step for single experiment runs. In the old architecture both steps have been coupled together, causing many users to wait for a compilation, while the physical experiment was in use by a long running job.
Our new implementation of the DCL middleware builds upon earlier research results in the area of service-oriented architectures, which are briefly described in the following section.

2 Stateful Service Concept

In order to realize the DCL as service-based distributed environment, we applied results from our earlier *Service Infrastructure* research [8] to the domain of remote/virtual laboratories. Our stateful service approach extends the idea of stateless Web services with the explicit notion of *service instances*. Client applications (such as workflow engines in typical SOA environments) are enforced to perform an explicit service instantiation through a factory operation. The factory returns a reference to a *logical service instance*, which is described as *WS-Addressing*-compliant XML document [2]. This document is then used by the client for subsequent service invocations, which are all automatically related the initially created session between client and server.
A logical service instance represents a stateful entity to the client, but does not necessarily need to be realized by only one *physical service instance* on a particular server. This slightly extends the idea of standard Web service frameworks, where services are referenced by an endpoint URI for a particular service instance on a particular machine. Instead, all clients communicate with an *coordination layer* that routes SOAP requests (specifically the SOAP body) to a matching execution host. All logical and physical instances relate to their according service implementation, which is realized as binary Web service component, such as a Java Servlet or a .NET Assembly. The kind of implementation for the particular service is transparent to the client in this case, and depends only on the available kind of execution hosts.
In our stateful service concept, a logical service instance has query-able state and monitoring information, expressed by uniformly accessible attributes. Our

implementation uses the specifications for the *Web Services Resource Framework (WSRF)* to allow interaction with stateful SOAP implementations. WSRF combines the *WS-ResourceProperties (WS-RP)* specification, which defines read, write and list operations for Web service attributes [1], and the *WS-ResourceLifetime (WS-RL)* specification, which defines operations and WS-RP attributes for managing the lifetime of a service instance [7]. Since both the attribute access and the lifetime management is independent from the particular service implementation, clients can access and utilize this functionalities in all cases. The according query and update operations become automatically part of the service interface, as defined in the according standards.

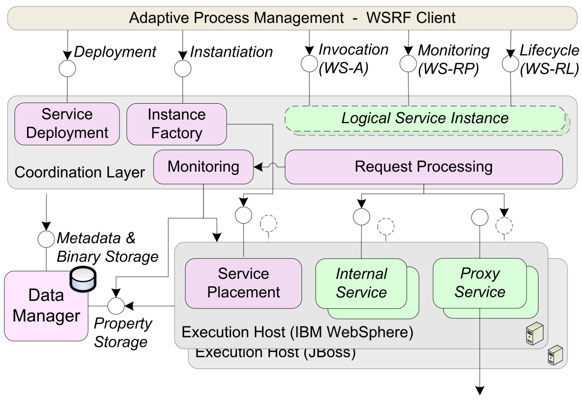

Figure 2. Service Infrastructure

A first implementation of this infrastructure model was realized and tested in the *Adaptive Services Grid* project for a period of 18 months. Figure 2 shows how the logical service instances for the client are managed by a coordination layer, which schedules and manages the incoming requests for the available set of physical service instances. New service implementations can be deployed at runtime, in which case the coordination layer chooses the right execution host for the binary (*service placement*). Service access is monitored by the request processing components, which supports the unified gathering of monitoring information for all service types. Service implementations can use a central storage facility to provide attribute values to the client or to save their own state between invocations. The atomic services can either implement some functionality by them self or act as proxy for external functionality.

2.1 Experiments as a Service

The experience with the existing Service Infrastructure from the ASG project showed the relevance of some architectural patterns for the identified DCL problems:

- Due to the usage of standardized SOAP-based protocols, client and infrastructure implementation can rely on different platforms.
- Stateful interaction with services is made available to the client in an interoperable and standardized manner.
- For scalability and fail-over issues, new services can be deployed during runtime. They can utilize additional execution resources on demand, without effecting active requests and their clients.
- Clients and services are loosely coupled. Service access and state data access do not relate to a particular execution host.

In order to transfer the Service Infrastructure concepts to the virtual laboratory application scenario, we compared the DCL concepts for the old infrastructure and the stateful service concepts of the ASG infrastructure. The resulting mapping is shown in Table 1.
Each DCL experiment controller, the software component to execute jobs on physical experiment hardware, can be represented by an experiment execute service implementation. It provides the necessary interfaces to execute jobs and query job results. Since most of the experiments expect the source code of a control application as input, we also introduced the notion of a compile service. A compile service is specific to an experiment type (as the execute service) and transforms source code to an executable binary, which can be directly passed to an execution service of a given experiment type.
The decoupling of compilation and job execution improves the scalability of single experiment types. Execute services act as proxy for the physical hardware, and can therefore not be duplicated to multiple physical instances on multiple computers. In contrast, the compiler service acts as self-contained functional unit, and can be replicated over multiple execution hosts.
As described in Table 1, every job for an experiment type can be represented by creating a logical instance for an execute service. The mapping between logical instances and physical instances is managed by the coordination layer. Each client therefore can operate its own logical instance (or session) for an experiment. The central request processing module queues the incoming requests for the available physical instances.
The standardized support for service attributes allows a unified representation of job results. Each physical instance can store job results from the experiment run as attribute values. The infrastructure relates such saved attribute values to the logical instance triggering the operation, and stores the value and related logical service instance identifier in a central database. If the client now queries its logical instance for some current attribute value, the coordination layer can provide the latest data made available by the execution service or the compile service. This decouples write and read operations for experiment data, and decouples clients from particular execution hosts for compilation or experiment services.

Distributed Control Lab	Stateful Service Concept
Experiment controller daemon	Execute service implementation
Experiment compiler daemon	Compile service implementation
Experiment simulation daemon	Experiment service implementation
Job for an experiment type	Logical service instances of compiler and execute service
Compiling a job	Operation on logical instance of compiler service
Running a job	Operation on logical instance of execute service
Job results	Attributes on logical instance of execute service
List of all available experiments	List of all usable execute services
Status of users job	Attribute on logical service instance
Queue per physical experiment	Queue per physical service instance

Table 1. Mapping of DCL concepts on stateful services

In the following section, we will now describe the implementation of the updated DCL based on the stateful service concept.

3 Implementation Details

In the current implementation of the DCL infrastructure, new available experiments must be announced by providing an implementation of execute and compile service for a particular experiment type. The experiment provider uploads a *service package* as binary file, containing the service implementation and a *deployment descriptor* (see figure 3). The deployment descriptor contains meta-data such as scheduling configurations, a description of the experiment for the front-end display, properties of the service and the experiment type that is used to group compile and execute services. During the registration process, the WSDL of the compile and the execute service is augmented with the necessary operations defined by the WSRF standards for property and life-time management.

To use an experiment a logical service instance for a compile service and for a execute service has to be created by the front-end on behalf of the end user. The created instances allow the usage of an experiment by invoking the standardized `ExecuteExperiment` or `CompileExperiment` method on the logical instance. The DCL coordination layer ensures that a working physical service instance exists for any logical instances being successfully created. If necessary, it places a new service by loading the according service package to a remote host. Results of the experiment runs are centrally stored and can be accessed via the resource properties of the logical service instance. Experiment hardware cannot be shared among multiple jobs. Therefore, the coordination layer has to support the serialization of invocations for physical service instances.

Listing 1 shows a sample implementation of an execute service for the Lego NXT robot experiment. Users can write control programs for robot movement, and submit it to the infrastructure in order to see the resulting physical activities of the exper-

```
[WebService(Namespace="http://hpi-web.de/esvc")]
public class NxtExecuteService : WebService
{
  [WebMethod]
  public void ExecuteExperiment(byte[]
      experimentProgram)
  {
    WebCam.Record();

    NXTBluetooth.UploadProgram(experimentProgram);
    NXTBluetooth.StartProgram();

    // save experiment result
    PropertySupport.SetInstanceProperty
      ("Video", WebCam.GetVideo());
    PropertySupport.SetInstanceProperty
      ("DisplayOutput", NXTBluetooth.GetLog());
  }
}
```

Listing 1. Experiment implementation

iment hardware. In the implementation, the class `NxtExecuteService` derives from the `WebService` base class, indicating the implementation of a new Web Service. Each execute service must implement the `ExecuteExperiment` method, which receives the program image to be executed as binary array. The method is called by the coordination layer, based on the next request to be handled from the queue of pending logical instance calls.

As first step in the example implementation, a camera recording is started to save a video of the robot's movements during the job execution. Then the binary program image is transferred to the NXT via a Bluetooth connection. After the execution of the job, which is indicated over the Bluetooth connection, results of the experiments are saved in the infrastructure. The DCL implementation automatically determines the related logical instance being called and can therefore provide a generic attribute access library for experiment services. This simplifies the programming model, since the integrators of new experiments don't need to consider the logical instance handling of the coordination layer.

Listing 2 shows a shortened example for the implementation of a client using an experiment and fetching the results afterwards. In the listing, logical instances

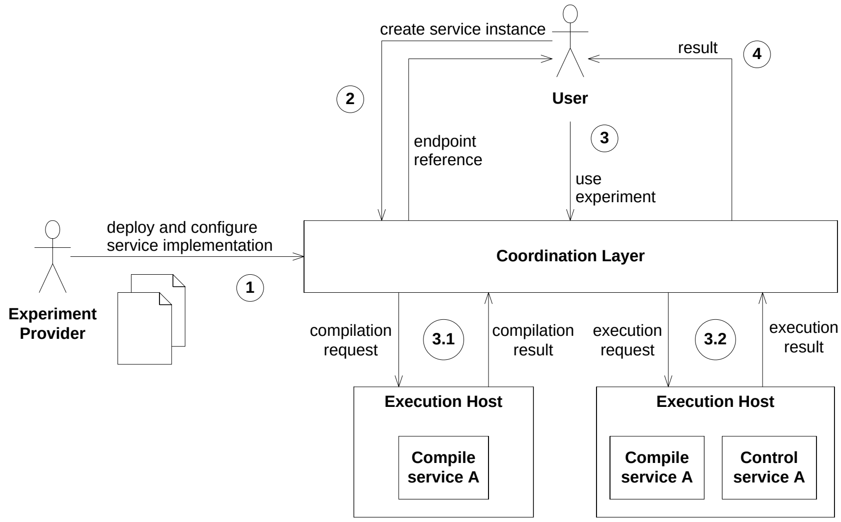

Figure 3. Workflow of activities in the Distributed Control Lab

```
// retrieve service instances
EndpointReference compileService =
  serviceFactory.CreateServiceInstance(
       CompileServiceId);
EndpointReference executeService =
  serviceFactory.CreateServiceInstance(
       ExecuteServiceId);

// compile code and execute experiment
byte[] compilationResult =
  compileService.CompileExperiment(code);
executeService.ExecuteExperiment(compilationResult)
     ;

// retrieve experiment result
byte[] experimentResult =
  executeService.GetResourceProperty("Result");
```

Listing 2. Experiment client

for compile and execute service are created first. Each service instance is represented by an endpoint reference object, in accordance to the WSRF specification. The `ExecuteExperiment` method returns after the finalization of the control program, or after the maximum time allowed for execution has been expired. The result of the experiment run can be acquired by accessing the according service attributes with the WS-RP operations.

All DCL experiments are currently accessible over the Internet. Therefore it must be ensured that only authorized users can access the experiments. The DCL therefore requires authentication data to be present in a SOAP request as described by the *WS-Security Username Token Profile* [3]. Furthermore, clients and experiment developers are free to use other WSS compliant mechanisms to protect the message body.

3.1 Scheduling

The classification and different treatment of incoming requests is another demand on virtual laboratory middleware, identified in the practical experiences with the classical DCL infrastructure. An open access to researchers, students and guest users at the same time demands a prioritization of specific requests according to the user identity.

In order to schedule Web service requests in our infrastructure according to an assigned priority, two problems needed to be solved. First, the priority decision value that is encoded in the SOAP message needs to be accessed. As most Web service stacks abstract from the communication handling, the access to priorities is usually not possible before the request processing starts. This prevents a re-ordering of incoming requests according to some priority setting. The problem was solved by intercepting the SOAP processing in the Web service stack, and analyzing the incoming raw XML data in a custom preprocessing handler. This step also covers the reaction on WSRF-compliant request messages, for example for the attribute access, which is not covered by the service implementation itself. The solution provides fast access to the priority values and allows the correct routing of the result messages.

Some of the experiment hardware requires a recovery phase between successive jobs. This is implemented by according queuing strategies in the coordination layer implementation.

Using our central scheduling approach, we are able to tolerate crash-faults of experiment controller machines, by having execution hosts installed on redundant computers. Before executing a job, the coordination layer checks whether the chosen physical service instance is still operational. If this is not the case, a physical service instance on another host is used. If no more physical instances are available, the coordination layer can also decide to deploy the service implementations to another empty machine. The infrastructure supports the addition of new execution hosts at run time, which allows the immediate reaction on failures without down time for the whole infrastructure.

In a future step, we plan to delegate jobs to multiple physical service instances in parallel and choose a result according to a voting mechanism. This mechanism is independent from the location of the execution host, and can therefore support fail-over scenarios between multiple interconnected virtual labs. For the sake of extensibility, the coordination layer itself is not aware of the deployment format of a service package. At the moment, our infrastructure contains two different types of service containers – one type to process .NET Web services, and one type for JAX-WS Web services. Since all incoming requests contain the information about the logical service instance, successive jobs need not to be processed by the same physical instance. This supports both load balancing and fault tolerance for a particular experiment type, under the assumption of reliable central data storage.

4 Operational Experiences

Within the VET-Trend project, we started a first pilot effort for integrating experiment installations at the Technical University Darmstadt and at the Hass-Plattner-Institute. TU Darmstadt is operating a remote laboratory for reconfigurable hardware modules, which can be programmed and tested by according tools. In order to perform the integration, TU Darmstadt is going to implement a service interface for their experiments, which will be used by our infrastructure. Beside the batch mode programming of hardware modules, we also identified the need for an interactive mode, which is required to perform test cycles at the downloaded hardware configuration. The interactive mode will realized as stream-based interaction with an experiment during the execution of a job.

Practical tests showed that the usage of SOAP messaging to query information about experiment runs is the most time consuming task in the new infrastructure. Since most of the job-related information remains constant during their life-time, we integrated several caches in parts of the infrastructure. With this technique, the number of fully processed service invocations at the coordination layer was dramatically reduced.

In the current implementation, we successfully interconnected an ASP.NET frontend with the coordination layer written in Java 6. Current execution host are programmed both in Java and .NET, and initial experiments already showed the possibility also for other platforms. In general, the usage of mature Web service standards and the consideration of WS-I regulations has shown to be helpful in order to achieve true interoperability in a heterogeneous middleware environment.

5 Conclusion

The utilization of service-oriented software architectures for remote/virtual laboratories is a promising approach for solving the typical problems of cross-organizational access, scalable behavior and dynamic resource usage. We presented our concept of an 'experiment as a service', were physical service instances provide access to the experiment hardware and logical service instances represent according user jobs. The application of mature Web service technologies allows to establish a transnational virtual laboratory environment, which integrates experiments and users from different sites all over Europe. First steps toward such an infrastructure already have been taken.
We have successfully used our laboratory infrastructure in courses on embedded systems, held at the Hasso-Plattner-Institute and the Blekinge Institute of Technology in Sweden. The integration of new experiments from other organizations has just started. Future work we will concentrate on the integration of further experiments, and on the improvement of our service-oriented laboratory middleware according to user and integrator feedback.

References

[1] S. Graham and J. Treadwell. Web Services Resource Properties 1.2 (WS-ResourceProperties). OASIS Open, June 2004.

[2] M. Gudgin, M. Hadley, and T. Rogers. Web Services Addressing 1.0 - Core. World Wide Web Consortium (W3C), May 2006.

[3] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker. Web Services Security UsernameToken Profile 1.1. OASIS Open, Feb. 2006.

[4] A. Rasche and A. Polze. Configuration and Dynamic Reconfiguration of Component-based Applications with Microsoft .NET. In *International Symposium on Object-oriented Real-time distributed Computing (ISORC)*, pages 164–171, May 2003.

[5] A. Rasche, M. Puhlmann, and A. Polze. Heterogeneous Adaptive Component-Based Applications with Adaptive.Net. In *International Symposium on Object-oriented Real-time distributed Computing (ISORC)*, pages 418–425, 2005.

[6] A. Rasche, P. Tröger, M. Dirska, and A. Polze. Foucault's Pendulum in the Distributed Control Lab. In *Procceedings of IEEE Workshop on Object-Oriented Realtime Dependable Systems*, pages 299–306, Oct. 2003.

[7] L. Srinivasan and T. Banks. Web Services Resource Lifetime 1.2 (WS-ResourceLifetime). OASIS Open, June 2004.

[8] P. Tröger, H. Meyer, I. Melzer, and M. Flehmig. Dynamic Provisioning and Monitoring of Stateful Services. In *Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST 2007)*, pages 434–438, Mar. 2007.

Service Orientation in Education

Intelligent Networks for eLearning / mLearning

Florin SANDU
"Transilvania" University of Brasov
Faculty of Electrical Engineering and Computer Science
Str. Politehnicii nr. 1, Brasov, 500024
ROMANIA
iesc.unitbv.ro
sandu@unitbv.ro

Titus Constantin BĂLAN
titus.balan@siemens.com

Silviu DOVĂNCESCU
silviu.dovancescu@siemens.com

Szilárd CSEREY
szilard.cserey@siemens.com

Siemens Program and System Engineering
Str. Mihail Kogalniceanu nr.21, Brasov,
500090 ROMANIA
www.pse.siemens.ro

Abstract

In Intelligent Networks (IN), processing of services is taken out of the switches and placed (as "middleware") in separated and distributed computer nodes. Thus, "service integrators" (and not only equipment manufacturers or network operators) can design, deploy and administrate telecom processes tailored to the specific needs of targeted customers (individuals or groups). Education is centered on communications, so IN can be used to enhance learning management systems (LMS), with no space boundaries and in total compliance with community goals. Proprietary IN platforms are used mostly by telecom operators, but open source platforms, using Java APIs (e.g. Mobicents) are nowadays available for educational service creation in universities – which represents the subject of this paper.

1 Introduction

IN can be used for implementation of a wide range of services that increase efficiency of the communication inside and between universities and enhance collaborative workflows. Before IN were standardized, service creation was mainly a role of specialized technical personnel. Since this type of personnel doesn't know directly the needs of end users, the service creation migrated towards the client zone, being available to "integrators" that are telecom sub-contractors (companies closer not to network operators but to end-customers).

Educational institutions can become such service integrators, creating specific IN services for education. Universities can act as service sub-contractors and even as service developers, using the great opportunity of open-source environments for service creation. Compared to overall number of end-subscribers of the telecom operators, the target user-group (consisting of students and teachers) is relatively reduced, so universities can also act as small network operators, using open Service Logic Execution Environments (SLEE) based on lower cost servers (see paragraph 4 in the following).

2 Intelligent Networks

2.1 Concept

IN is a concept valid for all kinds of computer and/or telecom networks, made as much as possible independent from the inferior layers in the OSI stack. IN integrate fixed and mobile telecom networks with computer networks and extend circuit switching and packet switching with the concept of service switching.

Intelligent networks enable flexible call control and new types of services.

Benefits of an IN:

- rapid development and introduction of new services
- rapid and efficient service customization

- independence of the Service Creation Environment (SCE) from the other parts of the IN
- open interfaces - providers can easily add new network elements for creating services, tailored to customer's needs

Typical IN Services:

- Number Translation Services (NTS): Freephone (FPH), Universal Access Number (UAN), Premium Rate / Tele-Info Services
- Mass Calling: Televoting (VOT)
- Corporate Services : Virtual Private Networks (VPN)
- Card / Mobility Services: Virtual Card Calling (VCC), Universal Personal Telecommunication (UPT), Prepaid (Card) Services (PPS), Personal Number Services (PNS)

These services are the most known and frequently implemented by mobile telecom operators. By using resource adapters that make possible IN working not only with GSM SS7 protocols, but also with other protocols, for example Internet streaming SIP (Session Initiation Protocol), RTSP (Real-Time Streaming Protocol) and by using open standards, the concept of IN is now extended and so is the area of services that can enhance eLearning and mLearning.

2.2 Architecture of an IN

IN produces the migration of service logic from switches to a database-centered system named SCP (Service Control Point). Still, the switching infrastructure is used for identifying the situations when communication with the SCP is necessary (see figure 1). That is the role of the SSP (Service Switching Points) that are SSS (Switching Subsystems) enhanced with detection of IN codes (e.g. dial-number prefixes like 0800), and triggering of SCP (that work as event-driven finite-state-machines – interrupt routines are namely the services to be invoked during a "call").

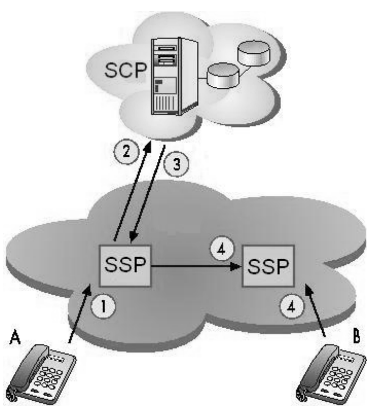

Figure 1: IN Architecture

Intelligent Peripherals are nodes which can connect to both SSP and SCP, in order to support the call with additional special resources (mostly related to voice data e.g. playing voice announcements or collecting DTMF tones from the individual end-users).

2.3 Service Creation Environment

From a user perspective, a service is a black box with external stimuli triggering the appropriate responses. This is an appropriate view for interfacing but it introduces difficulties when the implementation of a service is attempted.

An alternative viewpoint may be termed a “platform perspective”. This treats the service as a white box, exposing its internal operations.

Each step of the life cycle (see figure 2) will require contributions from different "stakeholders" and different parameters needed for the decisions. Each step can be seen as a use-case of the Service Oriented Architecture (SoA) with its own actors defining Business and IT requirements.

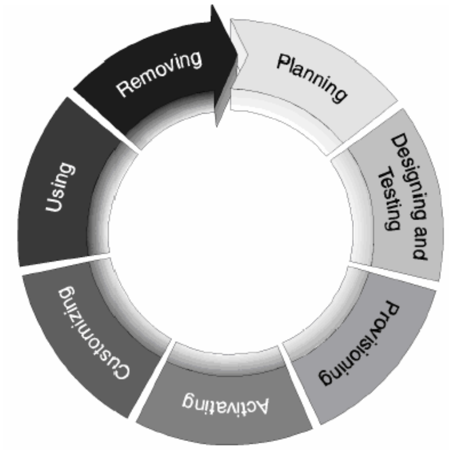

Figure 2: Service Life Cycle

The Service Creation Environment (SCE – see figure 3) is mainly a development environment: although the standards permit any type of environment, it is fairly rare to see low level languages like C used. Instead, proprietary graphical languages have been used to enable telecom engineers to create services directly. The languages usually belong to "fourth generation" programming languages. The dedicated Graphical User Interface (GUI) can be used to manipulate between different SIB (Service-Independent building Bloacks) as subroutines grouped (in a flow-diagram) to formulate a service (as FSL – Flexible Service Logic). The authors used Java APIs for Integrated Networks (JAIN) - one of the frequent technologies - in order to create IN services.

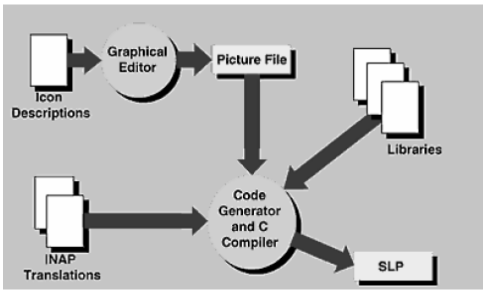

Figure 3: Architecture of a typical SCE

Roles of SCE are:

- creation of the service logic to perform the functions required
- defining the functionality of the service and the data model (type and scope of data)
- produce data management screens to manage the service and the subscribers to that service

3 IN services for education

3.1 IN Services that can be developed for eLearning and mLearning

The possibility of using multi-media streaming protocols like SIP and RTSP, combined with IN services, includes education in the large area of applications for intelligent networks. By innovative Voice- and Video- over IP streaming technologies new classes of services become possible.

First, IN can be used for *educational services that extrapolate the usual IN services* used in GSM telecom networks:

- **Call Center / Helpdesk / Hotline** and other online support facilities are mandatory to for modern learning environments.

The student-tutor relationship must be very interactive. Students often need fast problem solving and IN should help them to get almost real-time assistance. Since messaging and chat are very popular among student groups, Internet multi-media streaming and/or GSM can be implemented to access a helpdesk (where tutors or volunteer experienced students can offer assistance) with all IN routing: origin-dependent (for the closest / the cheapest connection), time-dependent (for different time-shifts and different working days, routing can be to local, zonal or national support centers) and, of course, the most common routing, which is destination-dependent.

Making the connection between the client, in this case the student, and the service provider - in this case the tutor (more generally, the "advisor" can be taken-over by experienced students) - is the role of IN.

Helpdesk services can have a dedicated IN number (e.g., 555-ITHelp) or "global-title" (e.g., in case of SIP, helpdesk@university.com). Interactive calls can be routed to any tutor "on-duty" (available on local /zonal/national or even international "online support" scheme). Students from any university can make calls (with no roaming problems in case of VoIP).

By building "helpdesk offices" for more than one university, and *competence centers* for collaborative didactical work and even for research, *trans-national networks* of universities can share not only information but also specialized support.

- A real advantage can be taken from the *charging schemes of IN* for possible incomes from a support (consultancy) offered to external partners (e.g. Industrial companies, administrative bodies etc.) based on a **Premium Rate** Services (somehow symmetrical to **Free Hotlines** and **Helpdesks**).
- **Televoting** services are also useful with various forms of group synchronization in education:
  - group decisions on timing for common sessions (lessons, workshops, exams)
  - rating of "educational objects" (courses, seminars, experiments) from a multi-choice offer and
  - feed-back (anonymous or not) for trainers
  - elections in the campus life ("eDemocracy").

IN tele-voting can be implemented with different calls (to different "global titles") for different choices, (that have different voting codes assigned). Result of the surveys can be determined also by the frequency of calls to a certain IN number.

- Campus **Emergency** Services – via traditional phone calls or by VoIP calls

IN can be also used for bigger Learning Management Platforms, e.g. the **"Virtual Classroom"** with:

- *registration* by calling a special SIP address, with allocation of a special *profile* (different access rights)
- *authentication* – by SIP address identification and access to a Virtual Private Network (VPN) s can be automatically created, profiles can be identified and workgroups rights created
- *charging* – using all IN mechanisms (like real-time "online charging" or CORBA charging etc.) on a "credit" base (in relationship with paid taxes, companies' sponsorship or state-granted scholarship)
- *access to "content-on-demand"* from repositories (resource centers) – servers that store (as digital libraries) "educational objects" identified / cataloged / discoverable according to SCORM (Shareable Content Object Reference Model)

- *individualized "learning paths"* – tailored by tutors and/or by students themselves (for optional modules' choice and assessment scheduling)
- *virtual "examination rooms"* (using secured video-conferencing – multiparty, with upload of presentations, channels' and ports' reservation for streaming over IP)
- *competence delegation* of teachers that can be "on duty" anywhere on the web and forward their tutoring and assessment rights (to any colleague that has the capability to answer the requests, as registered in "competence matrix" databases).

The IN should converge in eLearning / mLearning with newest *social networking*, in the *rich Internet* with *semantic webs* ("networks of information" – where content not connections are the priority).

3.2 IN Services for experiment-based eLearing and mLearning

IN can enhance education based on distributed and remote labs by "pervasive equipment availability". Equipment can be anywhere and published / invoked by Internet, by ISDN or any other telecom network type. IN SCP can host routing tables towards distributed test infrastructures that can be grouped in "colonies" (according to "ontologies") and dynamically refconfigured. Using one equipment or the other is transparent to the user (that asks for a service and not for an instrument - logical addressing, by middleware, replaces the physical addressing).

At "Transilvania" University, SOAP (Simple Object Access Protocol) was used to implement instrumentation services using BPEL4WS (Business Process Execution Language for Web Services).

Service routing by IN is, of course, destination dependent but also origin dependent: in distributed learning environments, similar services can be discovered in different places but, as instrumental services need fast response to stimuli for remotely operated equipment, IN can route to the closest point. Time-dependent routing is also decided upon scheduling (time-tables of availability stored in the IN SCP).

3.3 Cisco Approach of Services in Education

The Cisco Service-Oriented Network Architecture ("SONA") is an architectural framework that delivers business solutions to unify network-based services such as security, mobility, and location with the virtualization of IT resources. The Cisco SONA framework enables an organization to easily deploy identity, voice, collaboration, mobility, and security services across an IN – tailored to specific users, groups of users, or applications.

This type of services was used also for educational platforms by Cisco, implemented with Unified Communications Systems: the academic community - teachers and students – is interconnected trough a dedicated network, with advanced video telephony over IP, that integrates a whole range of services, from alarms and video announcements, to telepresence. Students benefit from the University's strong emphasis on experiment-based learning – gaining hands-on experience by participating in undergraduate research, studying abroad, working in cooperative programs, or becoming involved in the arts.

Cisco SONA has a service oriented architecture (SoA), but not in the pure sense of IN. So typical IN services are included, but SONA refers more to the infrastructure, being structured on three layers:

- Network Infrastructure Layer
- Interactive Services Layer that takes care of Service Management and Service Virtualization: this layer takes care of the application delivery issues (security, mobility, storage services) and application-oriented networking (voice and collaboration and identity services)
- Middleware and Application Platforms, the base for the Application Layer and the Collaboration Layer

Key attributes of a SONA network include:

- Standardization, which increases asset efficiency by lowering operational costs to support the same number of assets.
- Virtualization, which optimizes use of assets, so that extra physical resources can be logically segmented to be used across distributed departments.

4 Building an IN and Service Creation using open platforms

4.1 JAIN

JAIN is part of a general trend to open-up service creation. Services can be created independently from the IN machine; their range is enlarged because more people are involved in service creation, as it is an open procedure. JAIN is a Sun-invented acronym for "Java APIs for integrated networks." JAIN aims for an enabling set of Java APIs to develop and deploy service-driven network applications. JAIN is an industry framework designed and specified by groups of industry partners and experts. The JAIN effort has produced APIs, ranging from Java APIs for specific network protocols, such as SIP and TCAP (Transaction Capability Application Protocol), to more abstract APIs such as for call control and charging, and even including a non-Java effort for describing telephony services in XML.

A major goal of the JAIN APIs is to abstract the underlying networks, so that services can be developed independent of network technology.

4.2 Event-Oriented Programming and JSLEE

The Service Logic Execution Environment (SLEE) is a well known concept in the telecommunications industry. A SLEE is a high throughput, low latency event processing application environment. JSLEE is the Java standard for SLEE. JSLEE allows multi-vendor environments, even in the service layers of telecommunication providers. Being Java-based, the paradigm of "write once, run anywhere" is supported and allows portable standards-compliant applications. Besides, JSLEE introduces network abstraction by the means of *resource adaptors*. The JAIN SLEE specification permits popular protocol stacks such as SIP to be connected as resource adapters. JSLEE is a solid foundation with a robust component model (see figure 4) and scalability characteristics designed for high-volume and low-latency signaling. JSLEE is designed for stringent requirements of core network signaling applications. The JSLEE specification is designed so that implementations can achieve scalability and availability through clustering architectures.

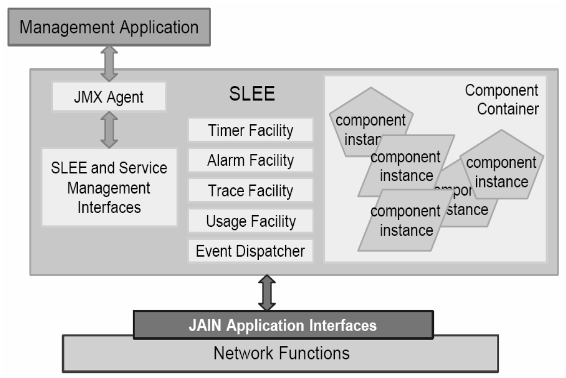

Figure 4: JSLEE architecture

The component container is the place where SIB-s (Service Independent building Blocks – see paragraphh 2.2) are developed. Java Management Extensions (JMX) is a Java technology that supplies tools for managing and monitoring applications, system objects, devices and service oriented networks.

The *resource adapters* represent software components that play a central role in the integration and connectivity between the IN and the other parts of the network. From this point of view, IN-s can be considered a point of integration between telecom networks and computer networks. Another part of the SLEE are standard facilities that help service creation: timers, alarms, event dispatchers, etc.

As already stated, implementation is based on events' handling. For example, using SIP resource adapters, if a SIP message is interpreted as encapsulating the address of a specific service, let's say emergency@university.com, this message is transformed into an event that is further linked to an application by the SCP (see figure 5).

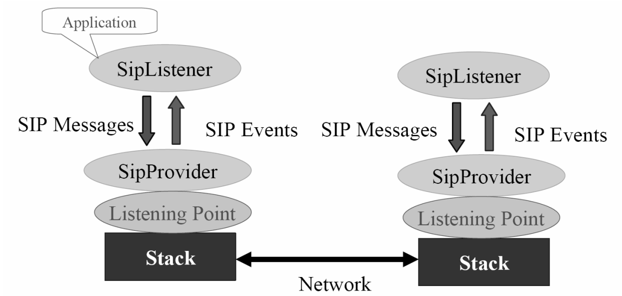

Figure 5: SIP communication based on events model

4.3 The Service Independent building Blocks

A Service Independent building Block (SIB) is a software component, a programmatic piece of a service. One can imagine them as LEGO pieces that make service creation very versatile and interactive. Adding a new service is much easier because it can be based on customization of multiple already created services.

A service instance can contain one or many instances of different types of SIB-s. The same SIB can be included in multiple Services but a single SIB can process only one event at a time. Multiple SIB-s belonging to the same Service can process events in parallel.

Typically, a root SIB represents a "complete service" (see figure 6).

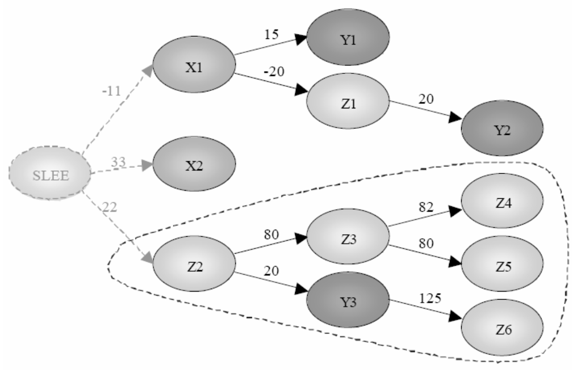

Figure 6: A SIB entity tree shows parent child relationship among instantiated SIB entities (X,Y,Z). Event delivery order is by priority, parent then child. Root SIB-s are specified in service.xml

For example, the SIB developer may develop a CallBlocking SIB and a CallForwarding SIB to implement the "call blocking" service and the "call forwarding" service.

These SIB-s are root SIB-s since they can be instantiated to block and forward calls, respectively. The SIB developer may create a new root CallBlockingAndForwarding SIB to implement the “call blocking and forwarding” service from the CallBlocking SIB and allForwarding SIB.

The SIB Developer distributes one or more Service deployment descriptors in a service.xml deployment descriptor file.

This descriptor file contains a single service.xml element. The service.xml element contains the following sub-elements:

- A description element - this is an optional informational element.
- One or more service elements.

Each of these elements is a Service’s deployment descriptor.

An example Service XML deployment descriptor file may be as follows:

```
<service-xml>
<service>
<description>
</description>
<service-name>FooService</service-name>
<service-vendor>com.foobar</service-vendor>
<service-version> 1.0 </service-version>
<root-sib>
<description> … </description>
<sib-name>Foo SIB</sib-name>
<sib-vendor>com.foobar</sib-vendor>
<sib-version> 1.0 </sib-version>
</root-sib>
<default-priority> … </default-priority>
<address-profile-table> … </address-profile-table>
</service>
</service-xml>
```

4.4 Mobicents implementation

Mobicents is the most popular Open Source SIP Application Server for the Java platform. It facilitates the implementation of new services in a simple way and quickly. Mobicents is distributed under GNU General Public License (GPL). Mobicents enables the composition of SIB-s such as call control, billing, user provisioning, administration, and presence sensitive features.

Mobicents servers are an easy choice for telecom Operations Support Systems (OSS) and Network Management Systems (NMS). In addition to telecom, Mobicents is suitable for a variety of problem domains demanding Event Driven Architecture (EDA) for high volume and low latency signaling. Mobicents fits-in as a high-performance core engine for Service Delivery Platforms (SDP) and IP Multimedia Systems (IMS). EclipSLEE is a graphical Service Creation Environment for rapid development of value added JAIN SLEE services.

Using the JAIN plug-in for Eclipse (see figure 7) one can obtain an already structured code, by much easier programming.

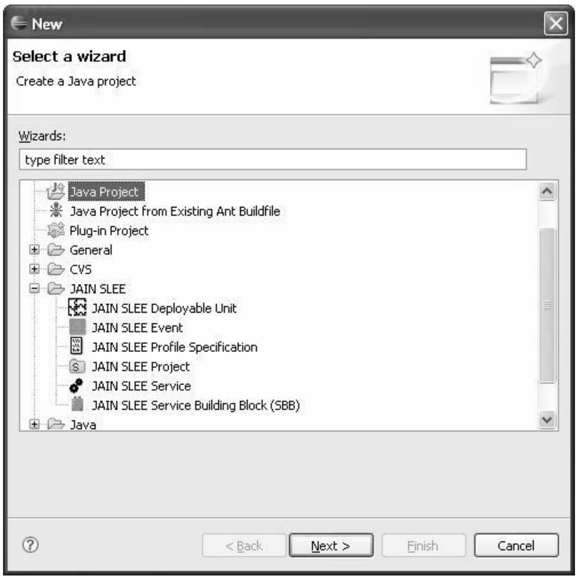

Figure 7: JAIN SLEE plug-in for Eclipse

Depending on the service implemented, the Java algorithms may not be so complicated; the simplest services may only redirect calls. But a SLEE is designed to serve a very big number of clients simultaneously, so this is why IN machines are powerful and expensive. But in an educational network, the number of users is much lower than in GSM telecom networks, so a simple server with a dual-core processor can act successfully act as an IN platform for eLearning / mLearning.

Our initial JAIN implementation, a SIP Helpdesk / Premium Rates service, using Mobicents as JSLEE, is a basic service. Of course, for the Premium Rates service, the charging service is also activated.

The database for the Service Control Point (SCP) is a MySQL database. For the Helpdesk service implementation, the database must have a redirect address. This address can have more alternates and can be changed by the IN interface to delegate, forward competence. Also charging information must be added in the database, and user information for security and identification.

Mobicents has no graphical interface, it’s just a server running in the background. The IN machine created at "Transilvania" University, in cooperation with Siemens Program and System Engineering is easy to use in the Internet. Users of the helpdesk must use a SIP phone (there are a lot of this kind of applications that are free to download, for example X-Lite) and must only call a generic SIP address, like helpdesk@unitbv.ro. The call is redirected by the IN machine to the helpdesk advisor (see figure 8).

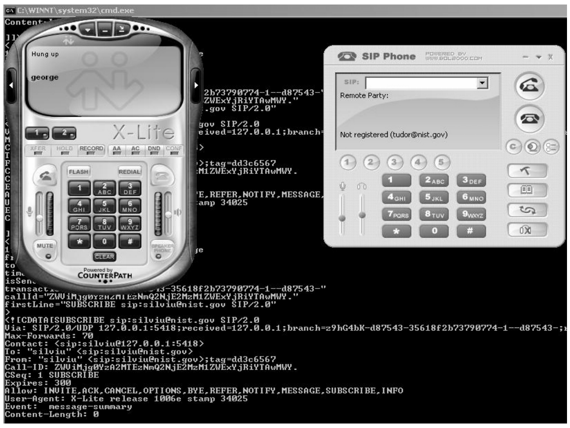

Figure 8: Simple mode of using the IN Helpdesk using SIP phones. The Mobicents server is running in the background

The advisor can be anywhere or can forward the call to another colleague - everything is transparent for the caller. The helpdesk system can be accessed from everywhere in the Internet, thus enhancing collaboration between research centers and universities.

4.5 Conclusions and further development

There are a number of learning concepts that can be implemented using intelligent networks.

The authors used the experience accumulated at Siemens, as an important telecom provider of IN systems, from the perspective of IN administration, but also from the perspective of the IN service creation.

Today's telecommunications migrate from big proprietary platforms, with dedicated hardware, to simpler platforms with standard interfaces but with more complex and specialized software (the “softswitch” concept). This trend is also taken into consideration by traditional IN vendors like Siemens. Using JAIN and JSLEE is a modern and inexpensive way to create IN-s and universities will certainly take benefit.

JAIN is extending its domain and also new resource adapters are now released for JSLEE. So JAIN will be used not only for IN platforms, but for a whole range of telecom systems. The new resource adaptors can transform common servers into IN machines but also into "media gateways". Using Java APIs for telecom platforms in education, servers could integrate streams of different types (in a working model similar with Cisco SONA) and take education into a new age of multimedia streaming.

References

[1] F. Sandu, D.M. Istrate - *Unified Benchmarks for Mobile Intelligent Networks* – Proceedings of the 36th International Symposium of the Romanian Military Equipment and Technologies Research Agency – “METRA” - 26-27 May 2005, Bucharest, ROMANIA, ISBN 973-0-03923-2

[2] https://mobicents.dev.java.net/

[3] http://www.mobicents.org

[4] JAIN SLEE 1.1 Specification, Public Review Draft

[5] Jepsen, Thomas C.; Anjum, Farooq; Bhar, Ravi Raj; Tait, Douglas (2001). *Java in Telecommunications: Solutions for Next Generation Networks* John Wiley & Sons, 2001

[6] Mueller, Steve. *APIs And Protocols For Convergent Network Services*. McGraw-Hill, 2002

[7] Henrik Berndt, Takeo Hamada, Peter Graubmann, *TINA: Its achievements and its future Directions*, IEEE Communications Surveys & Tutorials, First Quarter 2000, vol. 3 no. 1

[8] *Business Overview of Cisco Service-Oriented Network Architecture* http://www.cisco.com/en/US/netsol/ns477/networking_solutions_white_paper0900aecd803efff3.shtml

A Virtual Laboratory for Digital Design

Domenico Ponta
Department of Biophysical and Electronic Engineering,
University of Genoa
ponta@unige.it

Giuliano Donzellini
Department of Biophysical and Electronic Engineering,
University of Genoa
donzie@unige.it

Abstract

This paper presents a virtual laboratory for digital design, specifically conceived for educational application and for supporting project work both locally and at distance: the Deeds. It covers combinational and sequential logic networks, finite state machines, microcomputer interfacing and programming at assembly level. The virtual lab is based on three simulator integrated together, combined with a large repository of application projects, available on the web. The paper contains an example of a laboratory session structured accordingly to the Project Based Learning methodology.

1 Introduction

The rapid development of digital electronic techniques that has taken place in the last two decades has had a profound impact on the design practices, demanding a new educational approach in the preparation of the professionals working in this field.

The complexity of today's systems and the wide variety of their different applications dictates the necessity of using sophisticated software for conception, design, testing and manufacturing. Currently, Computer Aided Design (CAD) techniques are an essential part of the system development process, while traditional design and prototyping techniques have lost part of their role. Professional software tools are generally available for educational institutions and more and more often used in education, sometimes even in introductory bachelor courses.

In our opinion, the use of a professional commercial simulator in a course designed to provide the foundations of digital design is not a good choice. A tool conceived to increase the productivity of a digital designer does not meet, albeit as its primary target, the needs of education. What is a plus for the professional (for instance the wide availability of components and functions) may be a minus for the learner. The now common use of Hardware Description Languages for design is an handicap when the learner lacks programming experience and the high level of abstraction may hide important basic phenomena.

In reality, the complexity of today's digital systems comes out of the variety and complexity of the applications: per se, digital design is simple, provided that its founding elements are well understood and mastered. For instance, digital design is an almost ideal application field for Project (or Problem) Based Learning (PBL), since most of the training takes place during the development of projects, starting from only a small theoretical background.

The authors have explored the use of interactive multimedia materials as learning tools to facilitate understanding of the basic issues of digital design [1]. The pedagogical results suggested substituting most of them with a general purpose simulator. This is the rationale for the development of the Deeds (Digital Electronics Education and Design Suite), a virtual laboratory for educational application [2] [3].

2 The Deeds virtual laboratory

The Deeds has been developed at DIBE (University of Genoa, Italy) as part of its research activities in the fields of project based, distance and cooperative learning. It is extensively used by the students of the first and second year of electronic and information engineering, to support laboratory activity and project-bases courses.

The virtual laboratory is based on three simulators: the Digital Circuit Simulator (d-DcS), the Finite State Machine Simulator (d-FsM) and the Microcomputer Emulator (d-McE). They cover, respectively, combinational and sequential logic networks, finite state machine design, microcomputer interfacing and programming at assembly level.

The simulators can work together, allowing therefore design and simulation of complex networks including a mix of standard logic, state machines and embedded microcomputers, as today's applications demand.

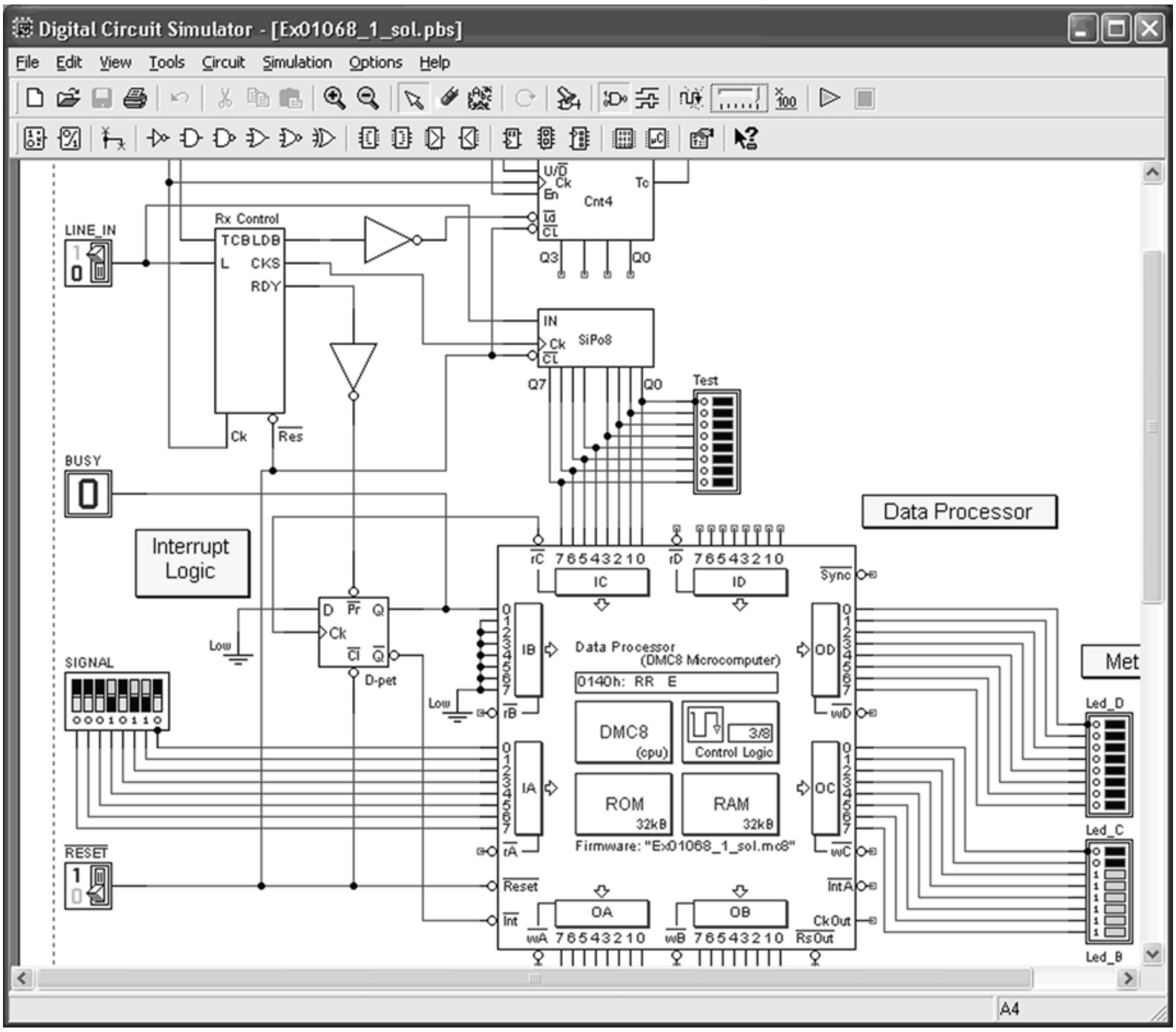

Figure 1: The schematic editor of the d-DcS.

The virtual laboratory is integrated with a Main and an Assistant HTML browsers: the former enabling Internet navigation to find pages with information, exercises and laboratory assignments, the latter providing step-by-step guidance to students in their work.

The d-DcS interfaces with the user through a graphical schematic editor (Fig. 1), providing a comprehensive library of logic components, implementing standard functions and not describing specific commercial products. The library includes also user-definable components, the user can define as Finite State Machines (FSM) and build with the Finite State Machine Simulator. The 8-bit microcomputer component is accessible through standard input-output parallel ports, besides other inputs as clock, reset and interrupt request: the firmware of the board can be programmed at assembly language level.

To draw the schematic of a circuit the student picks-up components from the toolbar, then connects them together using wires. Fig. 1 shows part of an embedded system based on the microcomputer and including library standard components, and a user defined one, designed as FSM. The user can test the network in two different modes.

In the simulation interactive mode, the student can "animate" the digital system in the editor, controlling its inputs and observing the results: such way of operation can be useful for the beginners.

In the timing mode, as in professional simulators, the behaviour of the circuit is shown in a timing diagram window, in which the user can define graphically an input signal sequence and observe the simulation results.

The d-FsM uses the ASM (Algorithmic State Machine) paradigm to graphically define the algorithm of finite state machines components (Fig. 2). The local functional simulation of the finite state machines provides the runtime display of the relations between state and timing evolution. These components can be directly used in the d-DcS and, also, exported as VHDL processes.

The d-McE is the tool for practicing embedded microcomputer programming and interfacing. The microcomputer component includes CPU, ROM and

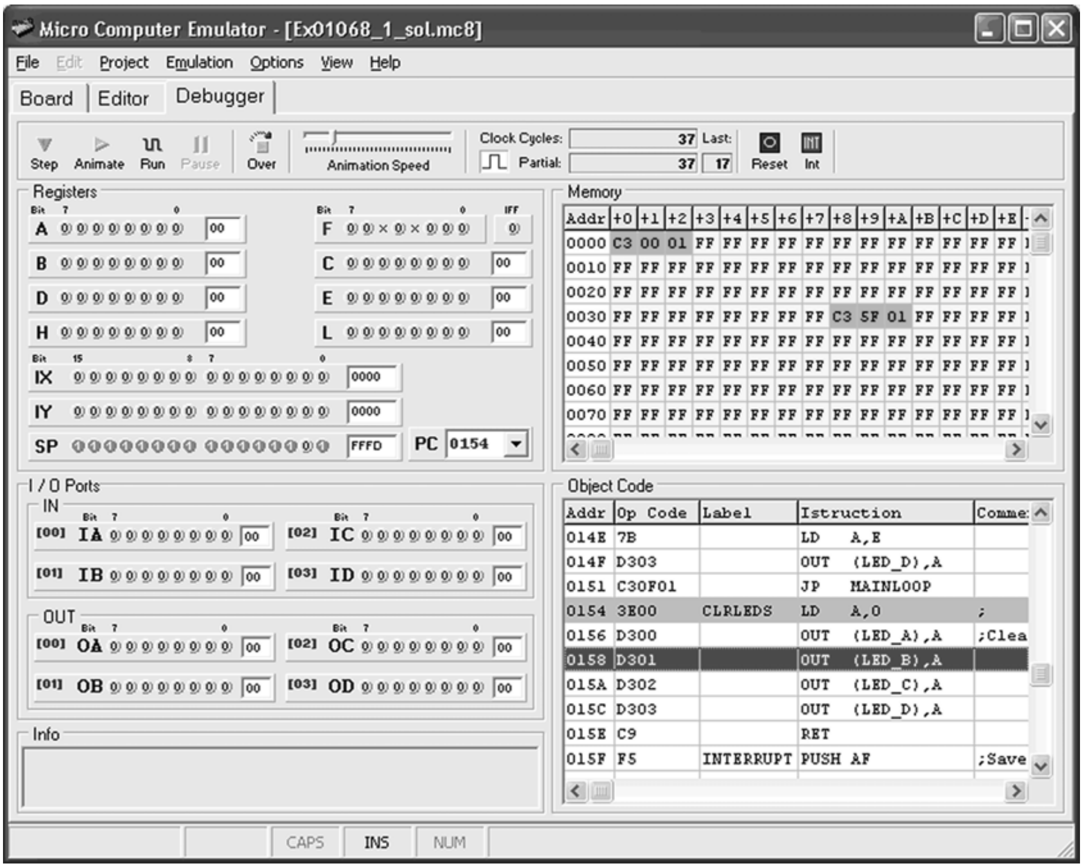

Figure 3: The interactive debugger of d-McE.

RAM memories, parallel I/O ports, reset circuitry and a simple interrupt logic. The custom 8-bit CPU (DMC8) is a simplified version of the ‘Z80’ processor. We have ruled out the possibility of emulating a state-of-the-art processor because we believe that the complex architecture is an obstacle to understanding the basic principles of machine-level programming.

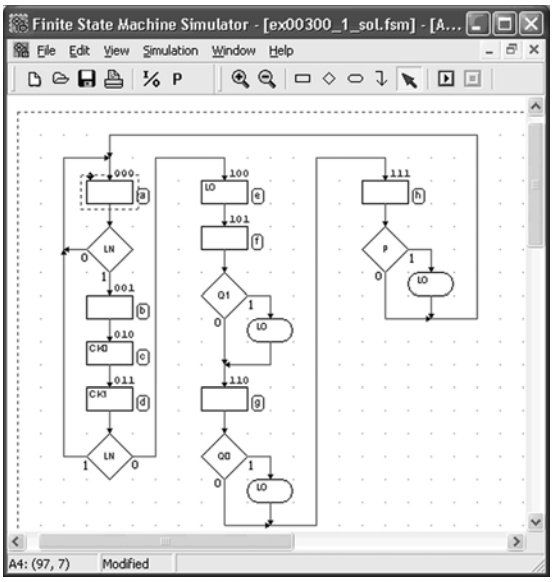

Figure 2: ASM chart in the d-FsM.

The source code editor enables user to enter programs, and a simple command assembles, links and loads them in the emulated system memory.

Programs can be executed step by step in the interactive debugger (Fig. 3), where the user can observe, at the same time, the contents present in all the structures involved in the hardware/software system.

In Fig. 3, the CPU register bank is placed on the top left and the Input/Output ports on the bottom left. On the right are visualized: the ROM and RAM memory contents (top) and the object and assembly code of the loaded program (bottom).

The instruction under execution is highlighted, as well as a breakpoint set by the user.

3 Pedagogical aspects of the virtual laboratory

In our environment, traditional lectures coexist with a problem-based laboratory, accessed either from a PC classroom, with tutorial assistance, or in distant mode. In both cases, the virtual laboratory is delivered as a resource of Moodle Learning Management System (LMS) [4].

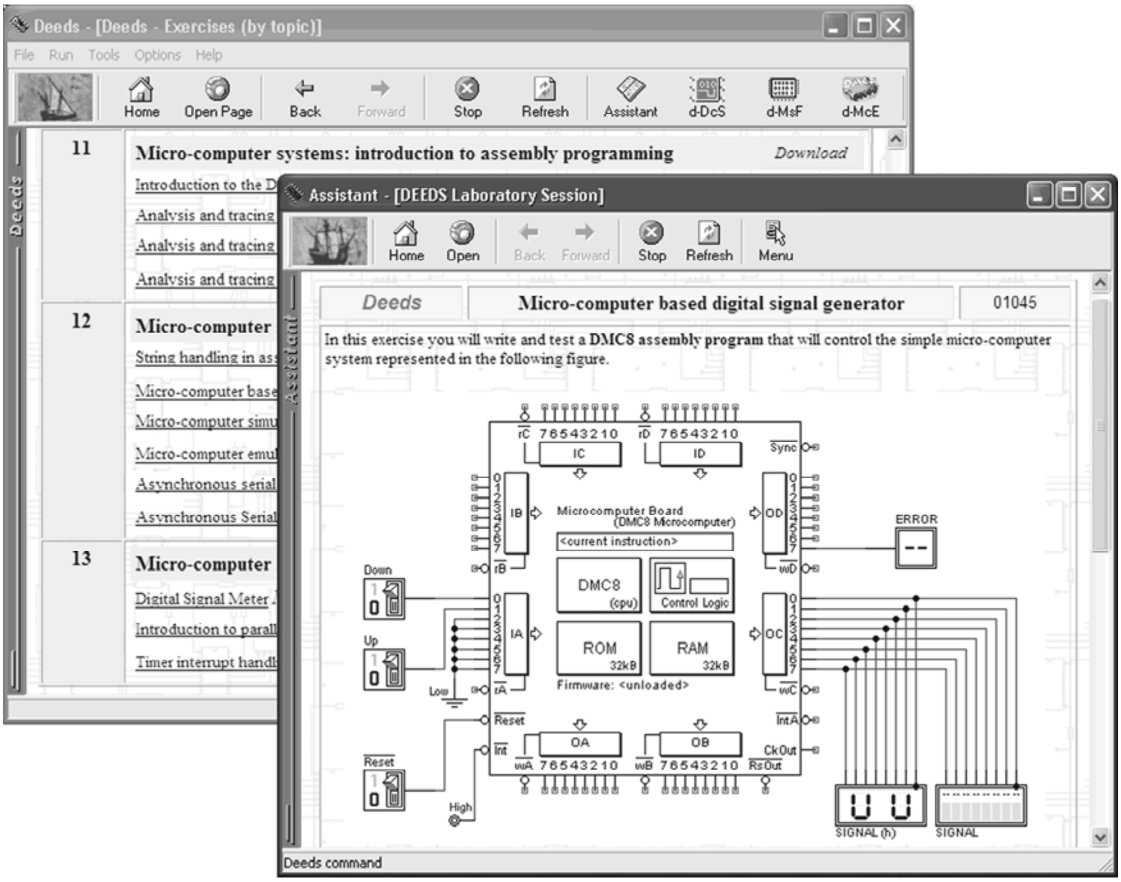

Figure 4: A virtual laboratory project assignment.

A typical virtual laboratory assignment is a project, presented as an HTML document with text and figures.

Fig. 4 shows a project assignment (foreground), chosen from the Deeds website repository (background). In this specific case the user is requested to design a digital signal generator by programming a given microcomputer hardware.

Text, figures and visual objects can be linked to the editing and simulation tools of Deeds. For example, let's suppose to assign a problem in the form of a schematic to analyse/modify. When the user clicks on the schematic, Deeds launches the corresponding simulator, and opens that schematic in it. As necessary, the Deeds open another browser (the Assistant) that can contain more information on how to design, explore or test the circuit itself. Such procedure is equally useful to convey concepts both on simple components and on more complex networks.

Project development phases may be guided by help and instructions supplied through the Assistant browser. This is mostly the case with introductory projects, where a step-by-step guidance may be in order. In more advanced projects the use of the simulation tools is less guided and left more to the user initiative.

An important features of Deeds is the ability to deliver a suitable trace of the solution (i.e. a partial schematic of the solution), or a set of stimulus signals, ready to be used to check the behaviour of the digital circuit under test. Using this approach, students can be guided to the desired level of problem solving, for example avoiding repetitive tasks.

A proper template for the project deliverables speeds up the preparation of the lab report, allowing the student to concentrate more in the technical work.

4 Example of a virtual lab session

Deeds lends itself very well as a tool for PBL. Learners download from the LMS the assignment, generally a functional description and a set of specifications of the system that they must design. We do not provide detailed instruction and explanations: instead, the project to develop is only an element of a set of problems, of different difficulty, joined together by the fact that they explore different facets of a main problem. The learner can move freely within the problems and choose if attacking directly the main one or reinforce his knowledge by approaching first the simpler ones.

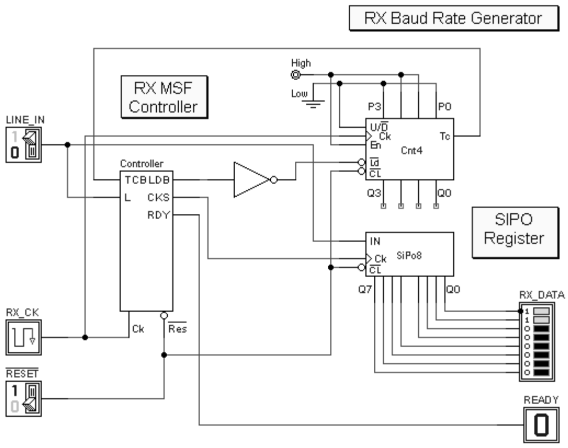

Figure 5: Asynchronous serial receiver architecture.

Serial data communication is a subject that allows to explore several aspects of digital design. As an example, therefore, we present here a problem having for target the design of an 8-bit asynchronous serial receiver.

Since this problems could be approached in different ways, assuming the student did not develop yet the capabilities to face the problem in an independent way, we propose as a common starting point the system architecture, of which the control unit must be designed (Fig. 5).

The set of specifications to satisfy, such as the format of the serial data, is submitted to the student.

Using Deeds, the controller can be designed as FSM. A possible solution to the problem is visible in Fig. 6, as an ASM chart, composed of 21 state blocks (the rectangles) and a certain number of decisional blocks.

The learners test the controller's design, initially, with the Finite State Machine Simulator. Next, they inserts the controller in the given architecture, simulating the whole system with the Digital Circuit Simulator.

Fig. 7 shows a typical result in the form of timing diagram.

Usually, students wish to familiarize themselves with simpler problems, before solving the complete problem. The problem of Fig. 8 asks for the design of a synchronous serial receiver, whose architecture has been simplified by removing the sampling clock generator. The serial signal protocol is the same, but the system now uses a sampling clock coming from outside.

As in the previous case, the students' task is the design of the control algorithm. It must be noted,

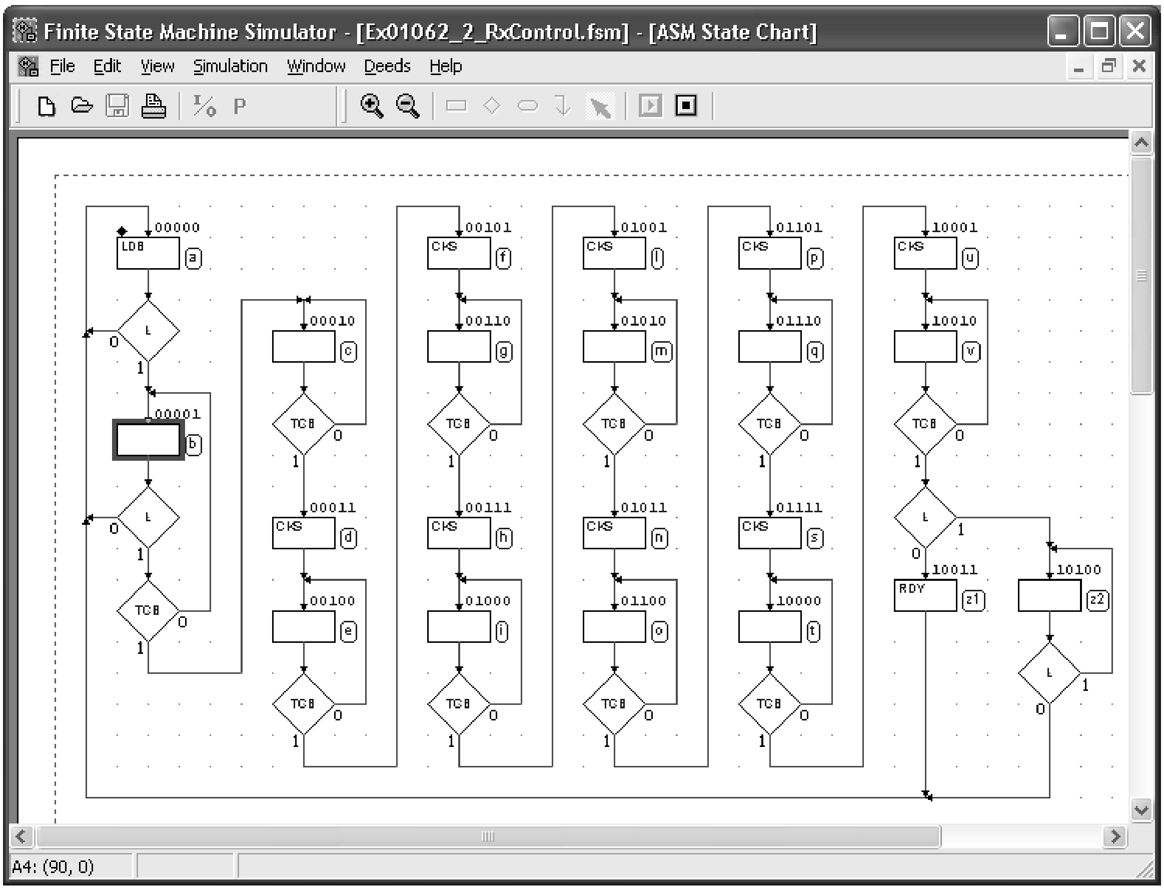

Figure 6: ASM chart of the synchronous serial receiver controller.

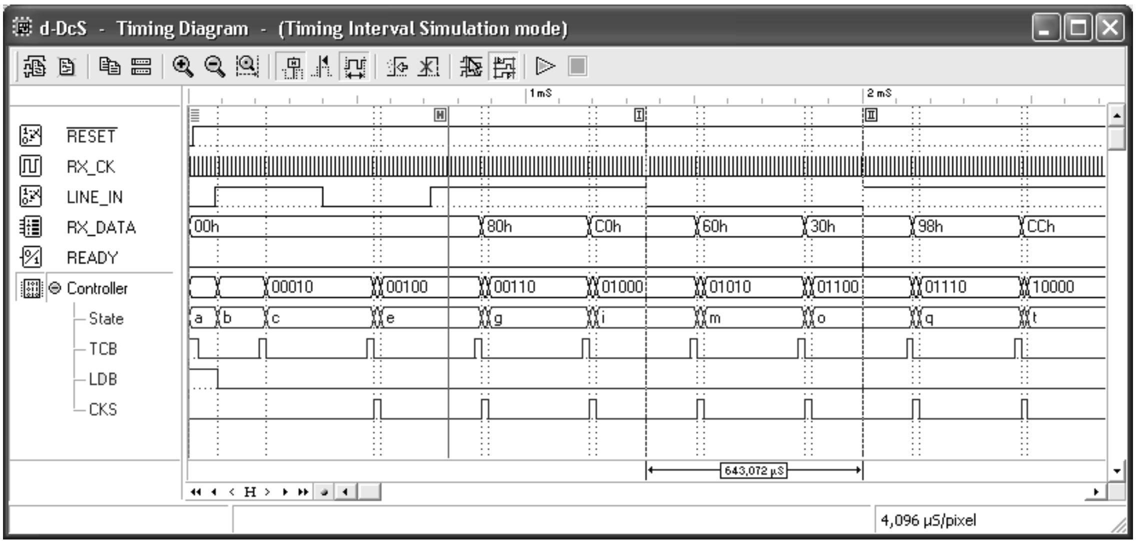

Figure 7: d-DcS timing simulation of the asynchronous serial receiver.

though, that the problem is not a subset of the previous one, but has different specifications, to avoid repetitions and keep attention alive.

Fig. 9 presents a possible FSM controller (the solutions to the problem are presented here to provide first-hand information on the level of difficulties).

In the pedagogical practice of PBL we do not publish any solution, expecting that students test by themselves the correctness of their design with Deeds's simulators. Correctness means that the final design operates accordingly to the initial specifications.

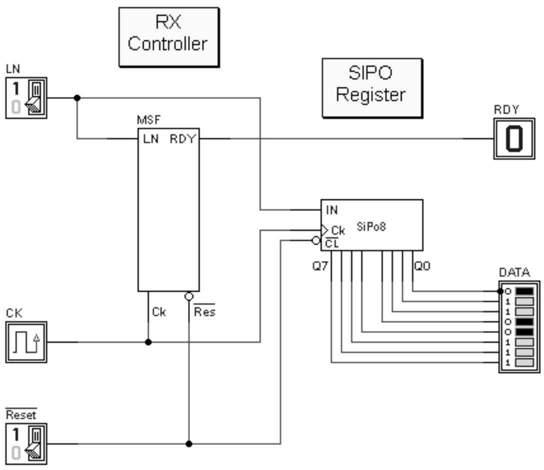

Figure 8: A simple synchronous serial receiver.

This target is achieved by successive design and testing phases. Pedagogically speaking, we are trying to remove the concept of "solution" when talking about digital projects. Not a solution but a design, in almost all cases not a unique one.

The role of the simulator is therefore central to our PBL approach, since it makes the learner autonomous in its work, removing the need to have somebody checking the projects.

The approach of providing a set of exercises of decreasing difficulty is quite flexible, since any problem may make a reference to a simpler one. It is possible, therefore, to build a complete course on the basis of PBL paradigm.

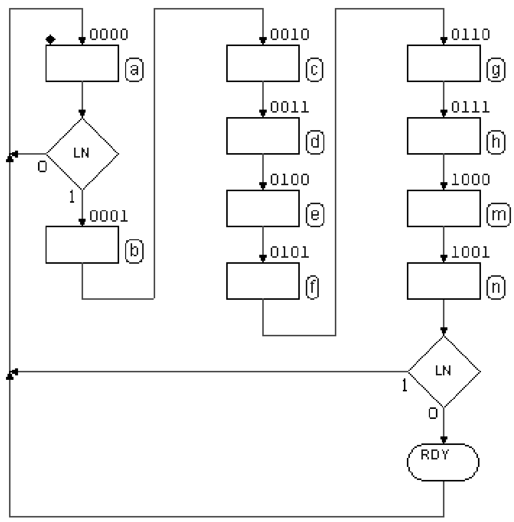

Figure 9: Synchronous serial receiver controller.

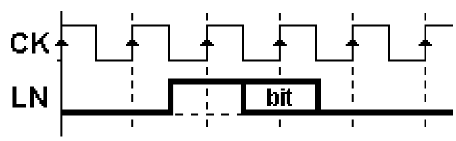

Figure 10: The very simple timing serial sequence of the "one bit" serial receiver.

As an example, the following entry-level problem ("one bit" serial receiver, Fig. 10) reduces the process of serial data acquisition to its simplest aspect, allowing the learner to concentrate on the principle.

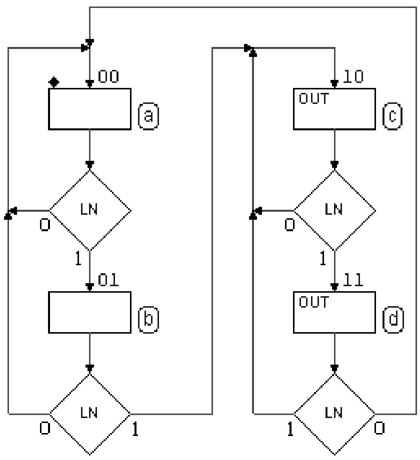

Figure 11: Simple "one bit" serial receiver.

In this case, the system does not need external components, by using directly inputs and outputs of the control unit. The design is represented by the FSM algorithm, shown in Fig. 11.

5 Conclusions

In a course supported by the virtual laboratory, acquisition of knowledge and skills takes place through the development of problems graduated by subjects and difficulty. A large repository of problems is already available [2].

Using the virtual laboratory, students can check by themselves their results and correct their mistakes, assuring, therefore, the self-consistence of their work, as it is the case of the professional designer.

The integration of the virtual laboratory with a Learning Management System provides added value for teachers and students alike. Teachers can keep track of students' activity, provide news and guidance, have access to the project deliverables. Students may carry on the project locally or remotely, can exchange information with their peers and get help by the teachers through the discussion fora.

At the level of instruction exemplified in this paper, the virtual laboratory may support or replace traditional hardware laboratories, providing several advantages, such as savings of time and resources, delocalisation, better integration with courses. The loss of contact with components, circuits and instruments is not, in our opinion, a vital issue, since the evolution of design techniques has made obsolete most of the traditional hardware experimentation and prototyping, replacing them with virtual and remotely operated laboratories [5, 6, 7, 8].

Deeds has been extensively and successfully used in our institution by thousands of students of the first and second year of the information engineering curricula, as a support to traditional teaching and effectively replacing traditional hardware laboratories. A thorough discussion of the tradeoffs between real and virtual laboratories in electronics is contained in [7].

Deeds has been specifically designed to support Project Based Learning not only in a local context but also in a geographically de-localized environment. Therefore it lends itself very well both for local and distance education and has the capabilities to extend project work within an inter-institutional and international context, as demonstrated by the results of NetPro (Network Based Project Learning) [9], a European project of the Leonardo DaVinci programme. Several colleagues from European universities have adopted Deeds in their teaching.

References

[1] G. Da Bormida, D. Ponta, G. Donzellini: Methodologies and Tools for Learning Digital Electronics, *IEEE Transactions on Education*, Vol. 40, No. 4, abstract p. 294, full paper on CD, 1997.

[2] Deeds website: http://www.esng.dibe.unige.it/netpro/Deeds/

[3] G. Donzellini, D. Ponta: A Simulation Environment for e-Learning in Digital Design. *Trans. on Industrial Electronics*, vol. 54, no. 6: 3078-3085, December 2007.

[4] Moodle website: www.moodle.org

[5] L. S. Indrusiak, M. Glesner, R. Reis: On the Evolution of Remote Laboratories for Prototyping Digital Electronic Systems. *Trans. on Industrial Electronics*, vol. 54, no. 6: 3069-3077, December 2007.

[6] L. Gomes: Programmable logic devices supporting embedded system design curriculum, in *Proc. 31st Annual Conference of IEEE*, IECON 2005, Industrial Electronics Society, paper on CD, 6-10 Nov. 2005.

[7] G. Donzellini, D. Ponta: The electronic laboratory: traditional, simulated or remote?, in *Advances on remote laboratories and e-learning experiences*, L. Gomes and J. Garcia-Zubia, Ed. Bilbao: University of Deusto, pp. 223–246, 2007.

[8] Jing Ma, J. V. Nickerson: Hands-on, simulated, and remote laboratories: A comparative literature review, *ACM Computing Surveys*, Vol. 38, No. 3, Article No. 7, pp 1-24, Sep. 2006.

[9] D. Ponta, G. Donzellini, H. Markkanen: Electronic system design: an experiment of project-based learning on network, *European Journal of Engineering Education*, Vol. 26, No. 4, pp. 375- 390, 2001.

Virtual and Remote Laboratories Feasibility Study and Development Guidelines

Vicent M. Rodrigo Peñarrocha, Javier Sanz López,
Bernardo Bernardo Clemente, Eva Antonino Daviu
ITEAM-DCOM, Universidad Politécnica de Valencia (UPV)
Camino de Vera s/n
46022 Valencia. Spain
vrodrigo@dcom.upv.es

Abstract

Virtual and remote laboratories are part of many e-learning initiatives. It takes a big effort for developping and mantaining. This effort should be taken into account when it is decided to develop a new virtual laboratory.The study presented in this article shows that many virtual and remote laboratories do not last more than two years. To avoid premature death of a virtual laboratory, some advice or guidelines are given, obtained as the conclusion from two surveys: one to the managers of many virtual laboratories, and the other to the students as the main users of these facilities. Educational institutions should fully support this new facilities.

1 Introduction

The big growth of internet technologies is changing the way of learning. The concept of distance learning is gaining space to traditional learning every day. Examples of evolution of distance learning con be found [1]. Engineering schools are joining this new way of teaching and learning, but they found one big problem: performing laboratory experiments without being present physically in the lab. This problem is the biggest obstacle for the distance learning, which has implied a lower quality level in comparison with the traditional learning.

Many solutions rely on simulations [2]. This is a good solution, because students can use and practice with a system very close to the real system. Here the concept of virtual instrument [3, 4, 5] is in many of the systems implemented, but it would be better if students could really make their measurements using the real instruments in the laboratory: controlling remotely the instruments and getting the results of the measurements done.

But the task of developing a virtual laboratory should be faced up the same way as any big project: Planning the initial development, the usual maintenance, the upgrades and the end of the project. This paper shows the results obtained from a virtual and remote laboratories feasibility study. This study has been developed as the first step to start the development of a virtual instrument that should be part of a remote controlled virtual laboratory that will be used by UPV students.

One of the main aspects to be discussed is the short life of many of the virtual laboratories developed, trying to reach the most complete conclusions about this situation. In order to get the data required for this study, a survey has been designed for the people in charge of virtual laboratories facilities, because they are the best experts. So these people have to be located and find the way to put in touch with them.

In many virtual and remote laboratories you can find different surveys asking about the user satisfaction when using the web site[6][7][8].

Other studies inquire about the difference between traditional classroom learning and distance learning [9][10].

This distance learning can be through old style (only books) or using new technologies.

Or you can find surveys about technology used in virtual laboratories development [11].

In this paper we use a survey to ask the people involved in virtual laboratories to know about the keys for success or give up in this task.

2 The survey to the virtual lab heads

2.1 Sampled population

During the first search step, near 50 web addresses and e-mails has been located to be the sampled population. Most of them are authors of different papers appearing in journals or conferences. Different search tools have been used and many references of multiple papers have been reviewed to locate the experts.

Also direct contact with experts in the UPV has been used to obtain the necessary information.

2.2 The survey

The e-mail has been the preferred option to contact the experts and ask for their collaboration.

In order to facilitate the response of the survey, with the aim of reducing the effort of those polled, it has been chosen to write the questions and possible answers in the way of a multiple choice answer, or with a brief answer. Going further in this objective of reducing effort, a web page has been developed, with the appearance of a form, in such a way that users do not even have to answer the e-mail. Only click on a link in the e-mail to access the web page with the survey. They have to click on the appropriate holes to answer, or fill small text boxes. The answers automatically are stored in the database.

2.3 The questions

All the questions were centered on six main aspects. Each one has been considered relevant due to different reasons that will be further discussed.

First of all there are a set of questions centered on the operating way of the laboratory, that's it: simulation, both on local machine and remote one, and remote control. It is interesting to know the differences and similarities between the different laboratories and if life expectancy differs depending on it..

It is also interesting to know which are the most used technologies, differentiating between the used for each one of the models of operation.

Next aspect has been the development process itself. It is interesting to know the development time, staff devoted to the development, the different stages of the process.

A very interesting aspect is the one related to funding of the whole project. It is tried to know the approximate costs of development and maintenance. Also the funding sources of the project, that can be from the developing entity or from different entities.

There are also questions about problems found during development and maintenance, in order to know real problems faced up by the developers and maintenance staff. This is a key factor because the development and maintenance mark the initial success and then, the life time of the laboratory.

The list of questions in the survey is:

1. Which actions are permitted by your virtual laboratory?
2. Which laboratory instruments are in your virtual laboratory?
3. Budget for the development of the laboratory. How the development has been funded?
4. How long took the development? Different stages in the development.
5. Staff involved in development.
6. Which are the tools and technologies used?
7. People in charge of maintenance. Monthly hours for this task.
8. Annual cost of maintenance.
9. Main problems found in the development.
10. Use of instruments is sequential or can be simultaneous?
11. How the control of the instrument users is developed?
12. What is the number of users? Frequency and period of use.
13. Is there any user satisfaction survey? Results?
14. An improve of the students results has been noticed?

2.4 The answers

From all the answers collected, here the most representative answers are presented.

Question 1: There is not a preference for the kind of virtual laboratory: simulation, local or remote, and remote control. Remote control systems use to be accompanied by complementary simulation systems.

Question 2: There are many different elements developed for virtual laboratories: instruments, robots, cameras,….

Question 3: The budget is very different, but all have money for the development. Not only the university but also external funding.

Question 4: The projects are complex so the development always is in stages.

Question 5: Most of the cases the development relies on grant holders. Different people in each stage, under the supervision of one or some professors. It works better when the person in charge of development lasts all time.

Question 6: Variety of tools and technologies, but Java, Matlab and Flash are the most repeated.

Question 7: Again depends on the project and budget: from 4 to 400 monthly hours spent in maintenance tasks. Set up the devices for different measures and creating and improving instruments are the usual tasks.

Question 8: From very little budget to numbers reaching 30 000 € per year, directly related to annual hours of maintenance.

Question 9: Looking for funds is the main problem. Programming hours for the development and maintenance requires money.

Questions 10 and 11: Most of them use sequential access to the instruments, not permitting various users share the instrument at the same time. In some cases simultaneous access is allowed with different monitoring systems.

Question 12: The number of users is reduced. Tele education institutions use it more intensively.

Question 13: More than half have done satisfaction surveys with positive answers in all cases. Students prefer firstly the real access to the lab. Second the remote control of virtual laboratories, and last, the simulation virtual laboratories. Some aspects can be highlighted: Students miss colleagues when working in the virtual lab, so a video chat utility or a virtual meeting point for laboratory could be appropriate.

Question 14: In all cases an improvement is stated, but less than expected.

3 The survey to final users

For a really useful virtual laboratory, final users should be satisfied. That's why in different years, a survey to users of the microwaves virtual lab in the UPV has been developed.

Main objective was to detect the problems noticed by the users in the services offered.

From the answers (76 students) to the survey, two were the main problems detected:

1. Lack of the traditional help: There was not a person near the user to answer their questions and doubts.
2. Slow response from the virtual instrument: time elapsed from button press to signal result in the screen.

So these two items should be addressed to help the final user be as comfortable as if it was in the real laboratory.

Of course many other issues should be addressed in the design of a virtual laboratory as in any e-learning project. Some of them can be found in [12]

4 Results analysis

In spite of the facilities for filling the survey, the answers only have reached the 15%. This low result has two origins:

1. Any survey has very low interest in being answered.
2. Many of the e-mails available in papers are obsolete or non existent, the same as the virtual laboratory related to these e-mails.

It does not depend at all (no correlation) between the laboratory model and the life time. Nor correlation between life time and technologies involved.

In simulation, Matlab is the most common programming language.

All surviving laboratories have a budget to develop and maintain them. Directly related the amount of the budget and the facilities offered by the virtual laboratory.

In universities, many laboratories grow up thanks to the enthusiasm of the professors involved, but past one or two years, when one or two papers in journals have been obtained, and with small or no financial help, the virtual laboratory vanishes.

5 Guidelines for a feasible virtual laboratory

Based on the answers received from the two surveys made to heads and users of virtual laboratories, and on the experience of authors, the next guidelines for a feasible virtual laboratory can be derived.

They can be classified in three aspects:

- Technological: all details directly related to technologies involved and technical aspects in the development of the project.
- Social: aspects related to human actions and human relations.
- Funding: money is needed to maintain a virtual laboratory.

Technological aspects:

1. Decide what kind of laboratory: simulation, remote control or both.

   Hint: can start with remote or simulation and later add the other.

   Hint: Better start with small applications. If all work, in future all can be expanded.
2. Decide about the technology.

   Hint: Specific tools for single operative systems or navigators, use to be more powerful than multi platform developments. But it shortens the users available to use it. Use multi platform tools (non dependant on operative system, computer, navigator,…). They will last more and more users could use it.

Hint: Avoid installation of programs in user computer. There could be as many problems as different computer configurations there are. If the user only needs a navigator, it will be the best.

3. Develop one instrument or experiment.

   Hint: Start with basic functions.

   Hint: When programming, think in future for possible multi user operation.

4. User trial.

   Hint: User can be the final users or a selected group of testers to check the virtual instrument or device.

5. Refinement.
6. Regular maintenance.
7. Upgrading

Social aspects:

It can be divided in two: related to users an related to managing and developing.

Related to users:

1. Virtual teacher.

   Hint: When teacher is available, use any possibility to live connect student and teachers: Instant messaging, voice/video. If not possible, use a forum or e-mail as an alternative.

2. Virtual laboratory meeting room.

   Hint: Students usually want colleagues to share experiences, ask,… while being in the lab. A chat room or common virtual place to meet, video/audio facilities or collaborative environments stimulate many student to improve results and can be an alternative to some easy questions.

Related to managing:

1. Management of the project.

   Hint: Usually one person is needed to dynamize the whole project. But more people is recommended to assure to take over when needed. If only one person is the really interested in the project, when he gets tired (or other higher priority tasks appear), the project vanishes.

   HINT: The best is to make the project an Institution project (university,….). as it assures future in management, daily work and best of all, financial support.

2. Motivation for daily work.

   Motivation for the managers and/or developers are different in time. Like true-life (and the results of the survey are very clear), the evolution is: First increases, then maintains, and in most of the cases, finally vanishes (Most of the virtual laboratories disappear past some years).

Funding aspects:

In all these stages, resources will be needed, so sources of funding should be found to achieve results.

1. Development
2. Maintenance
3. Upgrade

In this three stages the hint is the same and equal to last two items: The institution where the virtual laboratory has to be developed should be involved in the virtual laboratory to assure right development, maintenance, and the funds required for all aspects of the virtual laboratories.

6 Conclusions

As a new tool emerged from the e-learning technologies, virtual laboratories can help students access the laboratories in a new way: 24 hours a day, 365 days a year, from anywhere. This is an unprecedented possibility.

As a new facility, it requires a big effort to develop and maintain. This big effort becomes in a really young death of many of the virtual laboratories developed, making all the effort vain.

Trying to better understand this situation, two surveys have been developed and the analysis of the answers give guidelines for those who want to develop this kind of facilities. This guidelines center in three aspects: technical, social and financial.

And the final conclusion is: as a new facility, very useful for educational institutions, these institutions should support in all aspects the development and maintenance of virtual laboratories, the same way they support traditional laboratories, administrative services, institutional web pages and so on. If not doing this way, all the work depends on the mostly individual work of one, two or three people for each project that can end in any moment.

7 Acknowledgements

We want to acknowledge the reviewers for their helpful advice for this paper and for future work.

This work has been funded by the UPV during 2006 and 2007 under the PPI-06-05 program (primeros proyectos de investigación) of the Vicerrectorado de innovación y desarrollo.

8 References

[1] Hans-Henning Kappel, Burkhard Lehmann, Joachim Loeper, Distance Education at Conventional Universities in Germany, The International Review of Research in Open and Distance Learning, Vol 2, No 2 (2002).

[2] Gómez, J. M., Garcerá, E., Sanchis, A., Benlloch, J.V. Osciloscopio virtual: una herramienta didáctica accesible desde Internet, Congreso Nacional de Informática Educativa. (CONIED'99), 17_19 Noviembre 1999, Puertollano, Ciudad Real. Spain.

[3] Santori, M., An instrument that isn't really, IEEE Spectrum, August 1990, P 36-39.

[4] Spoelder, H. J. W., Virtual Instrumentation and Virtual Environments, IEEE Instrumentation & Measurement Magazine, September 1999, P 14-19.

[5] Goldberg, H., What is Virtual Instrumentation?, IEEE Instrumentation & Measurement Magazine, December 2000, pp. 10-13.

[6] Jian Qing Yu, David J. Brown, Ellen E. Billet, Development of a Virtual Laboratory Experiment for Biology, European Journal of Open and Distance Learning (EURODL).2005/II.

[7] Van Poppel, B.P. , Benson, M.J., Boettner, D.B., Arnas, A.O., Virtual laboratory development for undergraduate engineering courses. Information Technology Based Higher Education and Training, 2004. ITHET 2004, 31 May-2 June 2004, pages: 644- 649.

[8] Tonya Harris, Timothy Leaven, Paul Heidger, Clarence Kreiter, James Duncan, Fred Dick, Comparison of a virtual microscope laboratory to a regular microscope laboratory for teaching histology, The anatomical record, Volume 265, Issue 1, 2001.

[9] Thomas L. Russell's book, "The No Significant Difference Phenomenon" (2001, IDECC, fifth edition). http://nosignificantdifference.wcet.info/

[10] Carol A. Twigg, Innovations in Online Learning: Moving Beyond No Significant Difference. http://www.center.rpi.edu/Monographs/Innovations.html

[11] Nesimi Ertugrul,Towards Virtual Laboratories:a Survey of LabVIEW-based Teaching/Learning Tools and Future Trends*,International Journal of Engineering Education, Volume 16, Number 3, 2000 p. 171-180.

[12] Sergio L. Toral et al., Identificación de las variables de diseño de herramientas de e-learning mediante el uso de los mapas conceptuales, International Conference on Engineering and Computer Education. ICECE'05, Madrid, 13-16 Nov. 2005.

[13] Vicent Miquel Rodrigo Peñarrocha, Modelo de referencia de laboratorios virtuales y aplicación a sistemas de teleeducación, doctorate thesis, November 2003, Universidad Politécnica de Valencia.

A Training Need Analysis of Trainers in the Field Of Technical Vocational Education Training

Mihai ROMANCA
Transilvania University of Brasov,
500024, Politehnicii 1, Brasov - Romania
romanca@unitbv.ro

Paul BORZA
Transilvania University of Brasov,
500024, Politehnicii 1, Brasov - Romania
paul.borza@siemens.com

Sorin Aurel MORARU
Transilvania University of Brasov,
500024, Politehnicii 1, Brasov - Romania
smoraru@unitbv.ro

Abstract

The TNA (Training Need Analysis) was undertaken by the Transilvania University VET-Trend staff as a feedback from different organizations that expressed the interest to use the Vet-Trend network learning facilities, with the aim to guide VET-Trend network in designing training programmes that will meet real and actual training needs

1 Introduction

The aims of the training needs analysis are to find new criteria and methods for further training needs identification and analysis of trainers and teachers involved in vocational training [1]. The gathered information will be used by the VET-Trend management staff to attract other potential organization users, including producers of educational materials; and those organizations that initially expressed the interest in the participation. The main objectives of the TNA are:

- to understand the base level of existing knowledge about e-Learning and the current job position and duties held by the respondents.
- to gain information on participants' motivations for joining teaching or training session activities
- to assess the participant's ease of access to the Internet, with impact upon our design decisions concerning anticipated amounts of on-line work.
- to gain information on the participants' existing basic computer-based skills
- to identify the current training and development needs of trainers and teachers
- to identify future training needs
- to adapt the training pattern to the individual needs of the trainees
- to expand and valorise the existing remotely accessed experiment-based VET system
- to plan the pilot testing activities & training sequences, for teachers and trainers, in the use of the expanded and optimised existent e-Learning platform, in order to integrate their multi-media courses and laboratories, their assessment schemes and evaluation systems.

A questionnaire was developed based on some existing experience in other e-Learning programmes and TNAs [2]. The questionnaire was organized in the following main sections:

- SECTION A - About You.
- SECTION B - Using computers in your work.
- SECTION C - E-Learning experience and foreseen benefits of using VET TREND network facilities

The main criteria in developing the questionnaire was both to satisfy the aims and objectives, and to be reasonable short, and that to not take too much to complete by the target group of respondents [3].
The questionnaires were distributed directly or by post to 7 organizations interested to use e-Learning, and virtual-laboratory facilities of the network in enhancing the knowledge and skills.
It was intended that the total number of feedback questionnaire be 90. Not all the presumptive respondents reply on our request. Of the 90 names on lists supplied, 59 responses were obtained (65.56%).

2 Presentation of Results

In November we receive the entire 59 completed questionnaires. All the responses comes from higher education staff with the distribution from Fig.1.

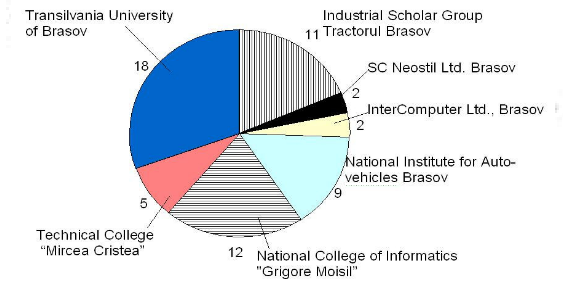

Figure 1: Distribution per institutions of questionnaire responses

We observe that there are a small number of responses from the two private companies. The first one (SC Neostill Ltd) is a textile production company and the number of employees that have access to a computer, to reach the e-Learning facilities of the Vet-Trend network is very low. The manager told us that at the moment he prefer to train the employees in by classical methods (face to face and learning by doing). The profile of the little company lead to the fact the from the five higher education staff (3 engineers, 2 programmers, 1 economist and 1 manager) only three has exclusive use of a computer with Internet access and the rest share another computer.

The other little firm (Intercomputer Ltd.) sell PCs and peripherals and self designed embedded systems. The bulk of the staff has higher education studies.

From the point of view of post held the distribution of respondents is indicated by Figure 2.

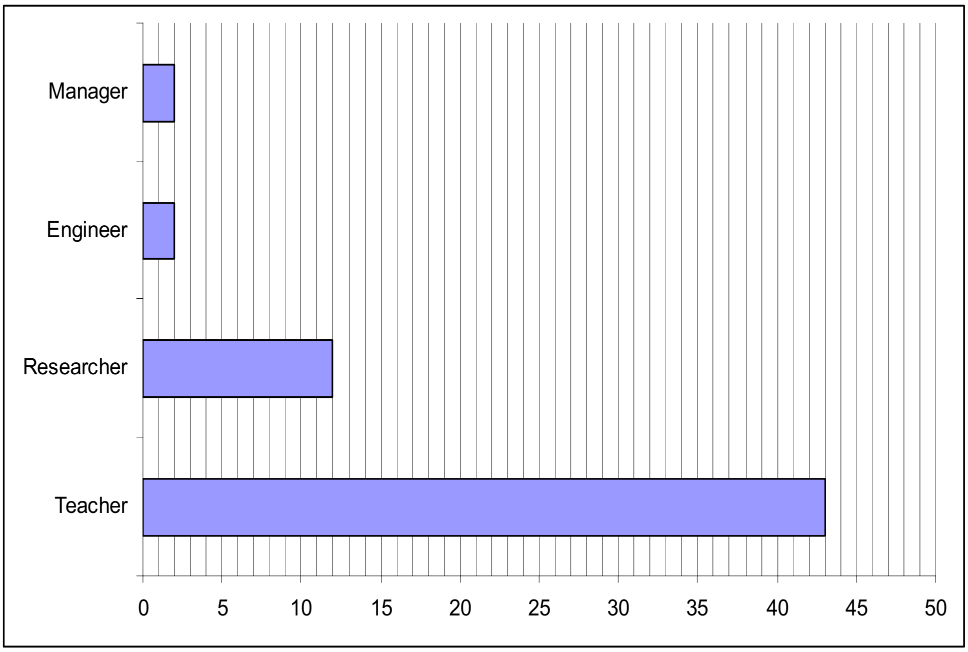

Figure 2: Current respondent's job

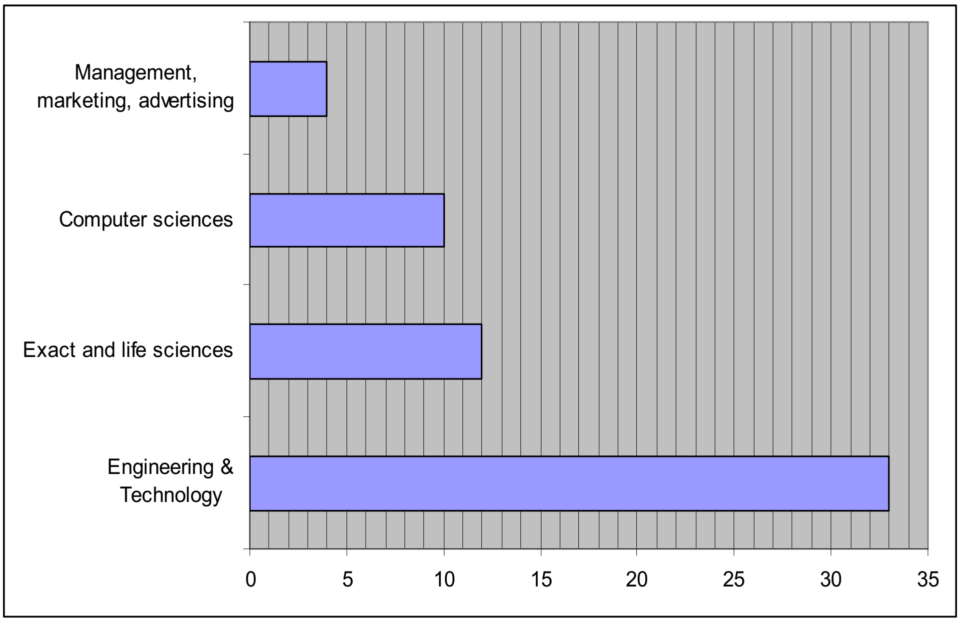

Figure 3: Main field of present activity

We observe that teachers and researchers mainly express the interest in learning and training with new methods, because for them the new skills are vital for their jobs. We think that the distribution is not very concluding because only two managers reply to our questionnaire.
The main field of present activity of respondents are graphically presented in Figure 3.
For the feedback questionnaire, the type of the institution in which the respondent's works is:

- University = 18 persons
- High school = 28 persons
- IT company = 2 persons
- Textile production private company = 2 persons
- Research Institute = 9 persons

At the question asking the direct access to a computer with Internet link the answers are synthesised in the diagram from Figure 4.
A lot of respondents use computer in their work to access and find information and resources as stated in the diagram from Figure 5.

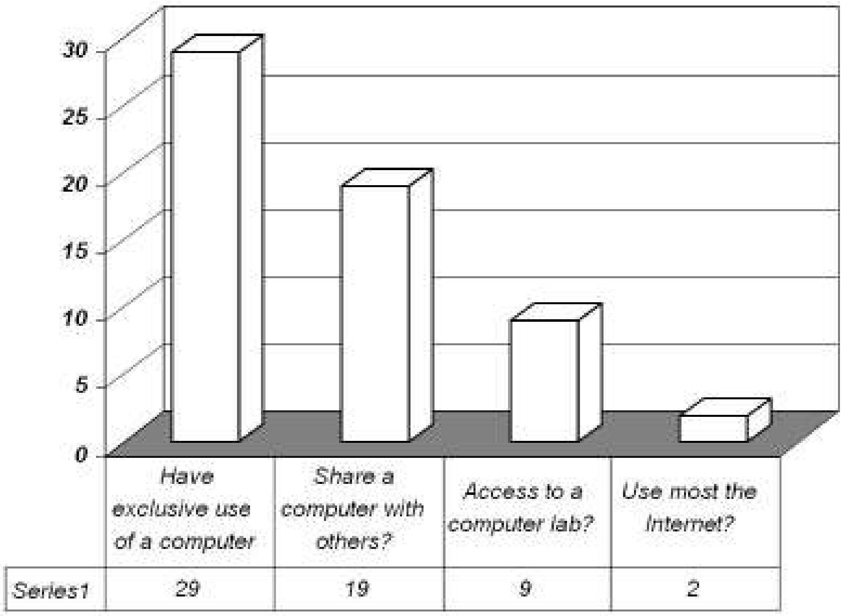

Figure 4: Access to computers and Internet

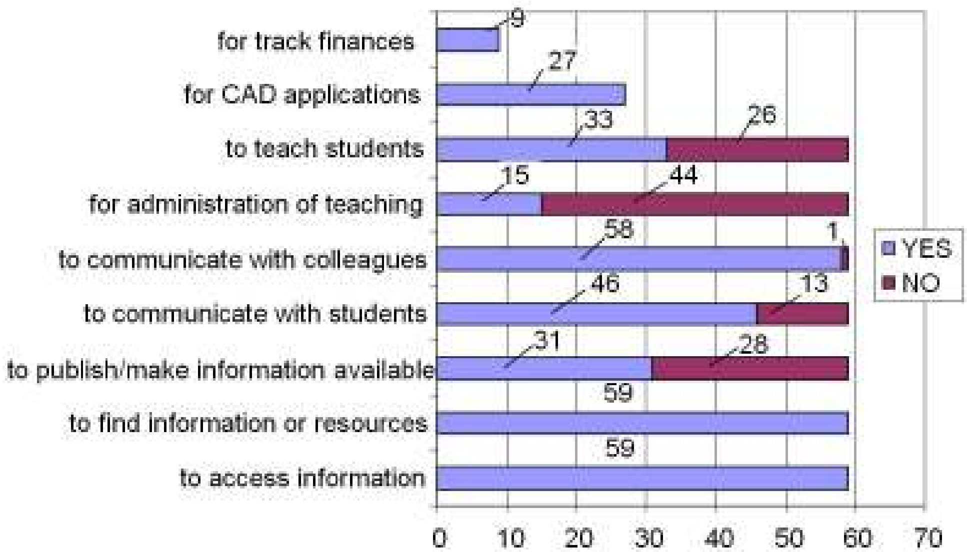

Figure 5: Mainly actual use of computer at work

About the e-learning experience 8 respondents declare to have incipient knowledge or cannot appreciate their experience (probably just hear about e-Learning), but 51 of the respondents (86%) declare to have a reach or sufficient e-learning experience and foreseen benefits of using VET Trend network facilities (Figure 6).

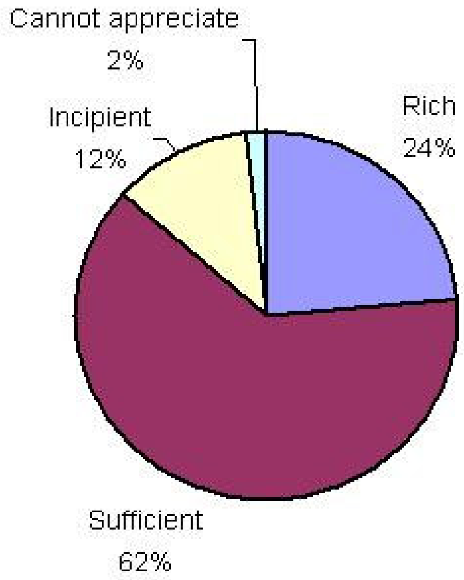

Figure 6: e-Learning experience

The future users of the network facilities will like to share experience in the domains indicated in Figure 7.

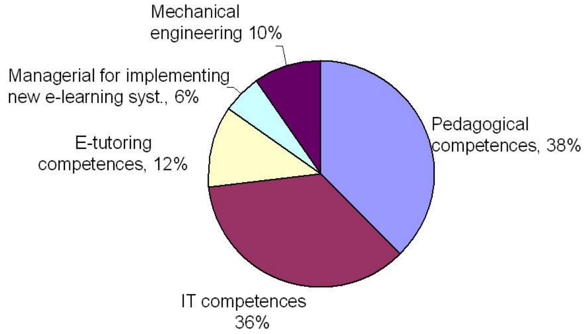

Figure 7: Competence areas

The foreseen benefits of the future users of the network based learning technology are presented in Figure 8.

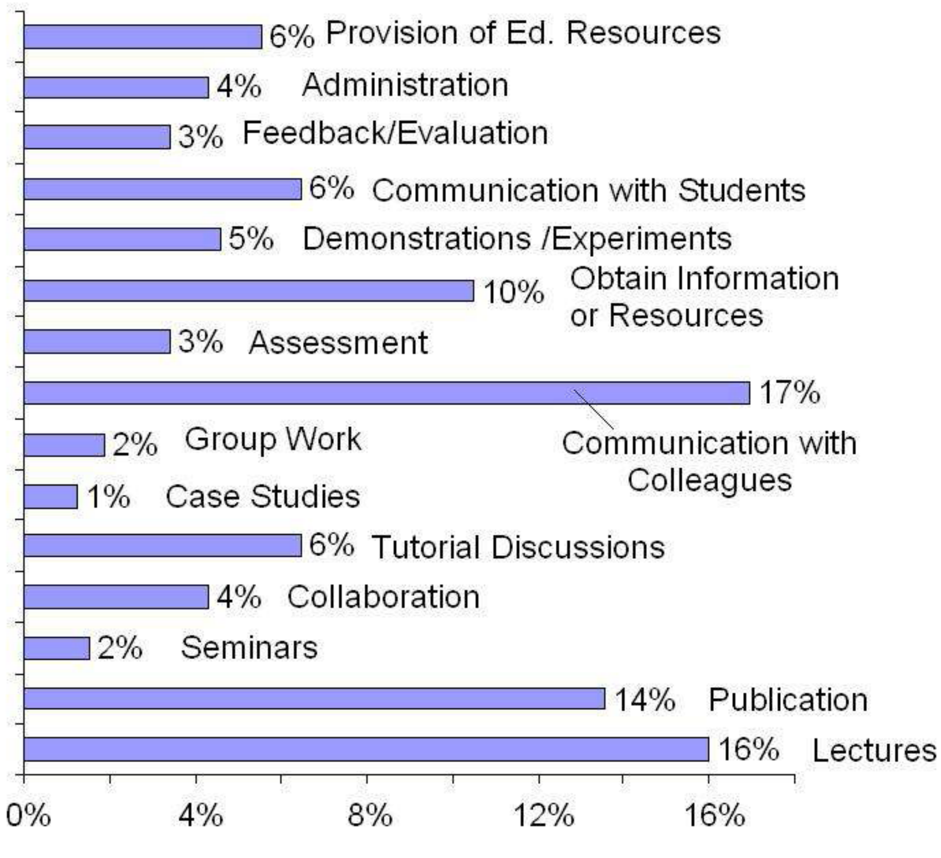

Figure 8: Foreseen benefits for the work-place

The most popular areas that the respondents see will have benefits at their work from using network based learning technology are: communication, lectures and publications.

In percentages, the benefits that the future learners wait are depicted in figure 9.

The respondents plan to will benefit in their work using the training techniques depicted in figure 10.

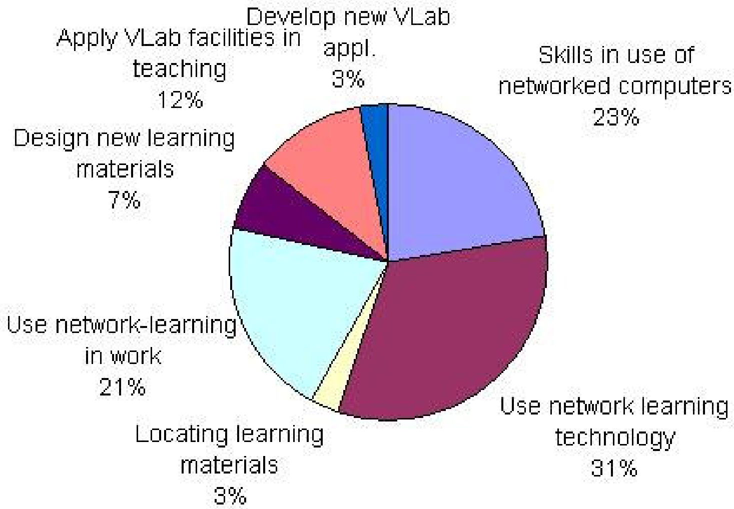

Figure 9: Foreseen benefits for trainees

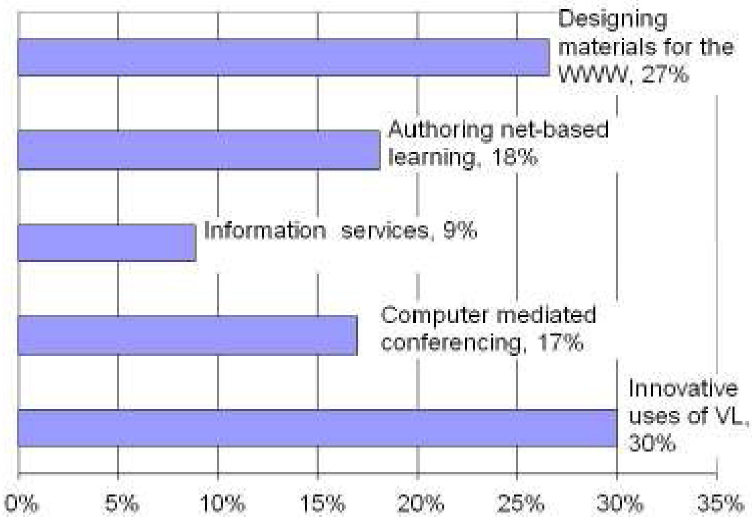

Figure 10: Training techniques to use

The most desired training subjects, useful in their work is synthesized in figure 11.

At the question that ask which of the some suggested resources and materials the respondents Which of the following resources and materials would you will use, the answers are presented in figure 12.

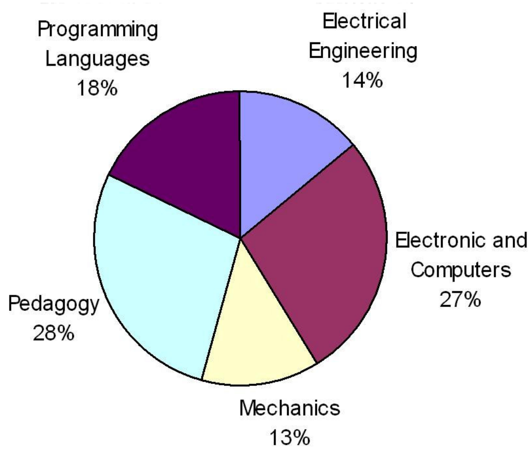

Figure 11: Desired training subjects

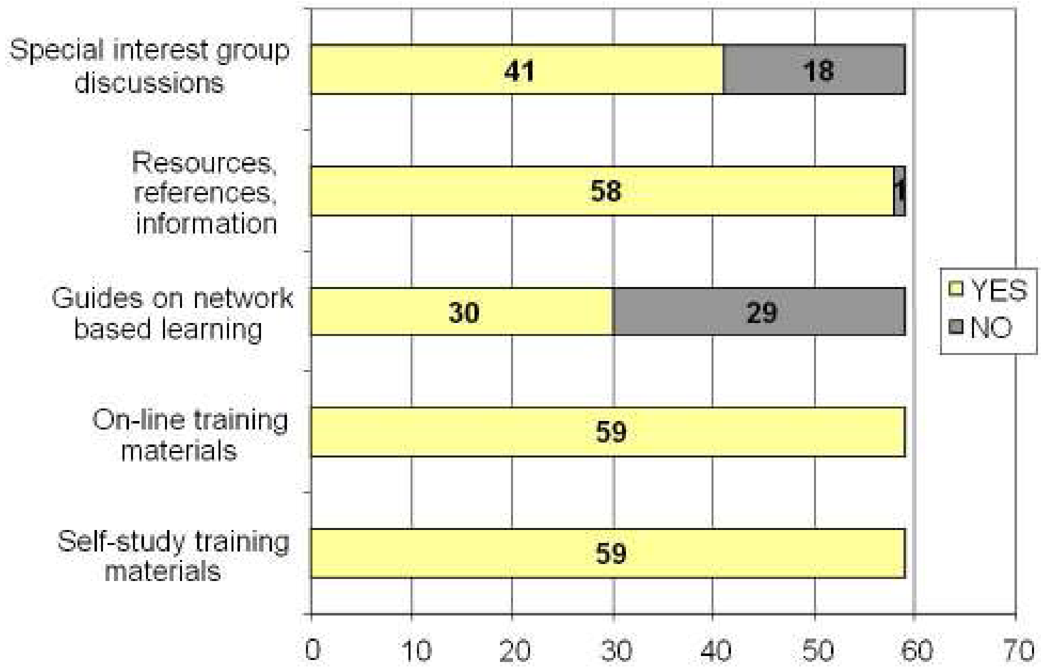

Figure 12: Resources and materials to use

At the question that ask what is the most convenient month of the year to undertake e-Learning activities through the network platform the responses are presented in table 1.

Table 1: Most convenient period for training

Month	Percentage
January	0%
February	0%
March	0%
April	0%
May	0%
June	32%
July	34%
August	7%
September	27%
October	0%
November	0%
December	0%

3 Conclusions

Reviewing the responses we can conclude that:

- A lot of the respondents use computers mostly to find and access information or to communicate with colleagues or students (Figure 5). There is a huge market for e-Learning activities, but a lot of target persons have a vague idea about the e-Learning offer, or do not find useful topic areas for enhance the knowledge and skill related to their job.

- In general teachers and trainers declare an e-Learning experience (Figure 6), but from the point of view of using the e-Learning technology the responses do not justify their responses. For example, according to Question C8 (*Which areas of your work would most benefit from using network based learning technology?*) and Figure 8, few of respondents see the usefulness of some important facilities of e-Learning systems: rapid feedback, on-line assessment (including self assessment), group work and group discussions, and the importance of case studies in learning / training activities.
- But, according to responses to question C.10. (*Which of the following training techniques would most benefit your work?*- with no more than 2 responses), and Figure 10, the majority of responses recognise the importance for training of using Virtual Laboratory experiments, and the necessity of designing adequate materials for e-Learning.
- The majority of respondents appreciate that the most desired training subjects are: Pedagogy, (28% of responses, we appreciate that the responses refer especially to e-Learning techniques, and teaching skills), Electronic and Computers subjects (27%), Programming languages (18%).
- In respect of the best period for training, the majority of responses prefer the holidays, Table 1, when their job permit to allocate sufficient time for these activities.

References

[1] Documents of the Leonardo da Vinci Programme *VET-TREND*, RO / 06 / B / F / NT175014

[2] Romanca, M., Neagoe, M., Cernat, M., WBT World – A web-based trainers' e-Learning tool, *Proceedings of the 3rd International Conference on interdisciplinarity in education ICIE'07*, March 15-17, 2007, Athens, Greece, ISBN 978-960-89028-4-8,

[3] Sarah Price, LOLA Training Needs Analysis, Institute for Computer-Based Learning, Heriot-Watt University, December 1998

Remote laboratories over different platforms and application fields: A Survey

Vasileios Kolias
National Technical University of Athens, School of Electrical and Computer Engineering, Athens, Zografou Campus
vkolias@medialab.ntua.gr

Ioannis Anagnostopoulos
University of the Aegean, Department of Inf. and Communications Systems Engineering, Karlovassi, Samos island,
janag@aegean.gr

Eleftherios Kayafas
National Technical University of Athens, School of Electrical and Computer Engineering, Athens, Zografou Campus
kayafas@cs.ntua.gr

Abstract

A survey over recent research in the area of remote laboratories is made, in the basis of the platforms used, the simulation environments and finally the application field of the conducted experiments. The survey is for the purposes of the V-Trend Program under the Leonardo Da Vinci framework and it is based on relative literature, which was mainly published or cited by international conferences, workshops, and periodicals (journals, magazines) mostly from the IEEE and ACM.

1 Introduction

In the last decade, the evolution of internet technologies opened up new prospects in distance learning. The wide use of internet enabled more users to use web-based teaching tools instead of more traditional teaching means. A significant improvement in communication within the academic communities rose along with the interest for the development of didactic applications that exploits these new potentials.

Laboratory experiments accompanying most of the courses, especially those that fall in the engineering domain, induce the need to be conducted remotely due to spatial, resource and time limitations. Apart from the hands-on experiments that can be done in real laboratories, two other types of laboratories exist: Simulated and Remote.

Hands-On Laboratories involve a physically real investigation process. Learners have to be physically present in the laboratory and they have to manually set up the equipment. According to [1], Hands-on labs provide experiences that clarify the disparity between theory and practice something in which the other types of labs lack. But the high demand of space, instructor time, and infrastructure that lead to high costs [2], along with the inability to meet some of the special needs of disabled students [3], make practice in hands on labs defective for the learners and inexpedient for the schools.

Simulated laboratories consist of computer software that imitates real experiments as if they happen on real infrastructure. Simulated labs are usually less expensive than Hands-on laboratories, less difficult to learn and operate and they are considered at least as effective as traditional hands-on labs because by fully controlling the simulated process they can review and understand it better [4]. But simulated laboratories lack of realism because the data from simulated experiments are not real and therefore learners cannot learn by trial-and-error [5]. Also the cost of simulated labs is not necessarily lower than that of real labs. The more realistic the simulation, the more time-consuming in development they are and therefore more expensive [6].

Remote laboratories provide affordable real experimental data through sharing experimental devices with a pool of schools [7][8]. They increase the number of learners that are able to conduct experiments along with the times they can conduct them. The ability to experiment from the learner's preferable place along with the idea of acting on real equipment makes the laboratories more effective to them [9]. But according to [10], remote labs are considered to be the same as simulated labs as concerning realism, regardless of the fact that they provide real data.

In this paper we present a survey of remote experiment applications in the last decade. Remote experiments can be grouped in two types: Remote control type and simulation type. The former utilizes a remote laboratory and therefore it uses actual equipment, yielding the same results with a hands-on laboratory. The latter utilizes a simulated laboratory, implying simulation software, which enables many students to perform experiments simultaneously, but lacks of realistic results.

We categorize those efforts according to the used platform, the type of experiments conducted, and therefore the area in which the experiments apply to.

2 Related environment for remote experiments

Due to the practical nature of engineering computer tools such as multimedia tools are considered less appropriate for engineering courses or courses that require hands-on experience. Their use is limited into courses which give individualized assistance to learners. However, recent advancements in internet technologies or applications based in LabVIEW have clearly shown that conventional experiments can be successfully integrated into custom written virtual instrumentation and can be delivered by computers and therefore be conducted remotely with ease.

Many interactive simulation, control and visualization software solutions that carry into effect the concept of remote experiments are available in the market today. NS2, Omnet and Matlab/Simulink for simulation experiments and LabVIEW for visualizing actual equipment are the most common environments in which the experiments are realized. However the diverse nature of the lectures and laboratory courses in each institution, often requires custom-build software. Java technology offers flexibility and ease of use for developing such software and is mostly preferred. When it comes to choosing the appropriate software to build virtual instrumentation, various criteria may be considered always according to the final goal [11]. The chosen software should allow to test individual modules easily and to develop applications quickly. It should enable the designers to work on separate parts and compile them on one platform. It should also allow incorporating with previous applications, and also with the previous versions of the software. It should be able to gather data from different interface hardware. The designer should be able to build libraries of low level routines to link them in higher level systems. It should also have advanced debugging features, to optimize product design and to determine a defect in the code. Executables should be provided to avoid alteration, to hide the code or to create standalone applications. Add-on packets should be provided to indicate the market acceptance of the product and speed the development. The end product should also meet the required performance. It should have an intuitive graphical user interface that enables a user to look at it and see what needs to be done. Finally it should have multimedia capabilities for future development.

2.1 LabVIEW

National Instruments' LabVIEW is the most frequently chosen software package because of its overall flexibility as an engineering tool [12]. This platform and development environment uses a dataflow programming language called G, which allows developers to create their own virtual instrument in a flexible, modular and economical way. Execution is determined by the structure of a graphical block diagram on which the programmer connects different function-nodes by drawing wires. These wires propagate variables and any node can execute as soon as all its input data becomes available. Most of the criteria mentioned above are reflected on this platform. LabVIEW supports extensively the access to instrumentation hardware. In terms of performance, it includes a compiler that produces native code for the CPU platform. The graphical code is translated into executable machine code by interpreting the syntax and by compilation. The fully object-oriented character of LabVIEW code allows code reuse without modifications. The LabVIEW Professional Development System allows creating stand-alone executables and the resultant executable can be distributed an unlimited number of times. The run-time engine and its libraries can be provided freely along with the executable. The nature of the G code is platform independent, making it portable between the different LabVIEW systems for different operating systems. Applications in the LabVIEW environment cover a wide range of engineering solutions varying from space technology to nuclear power area. According to the market and the users LabVIEW seems to be the primary choice in designing control and analysis solutions.

In a 3-tier architecture in [13] the LabVIEW environment is installed in a computer called Measurement Server which enables the interaction with one or more instruments. It is physically connected to a set of different electronic measurement instruments through an interface card. In the specific implementation each Measurement Server is connected to a single Laboratory Server, the only computer connected to the internet, in that way consisting a secure LAN.

In [14] the LabVIEW embedded dynamic web server that allows publishing and making available, through a web connection, the Virtual Instrument front panel in a straightforward way, is exploited in order to implement dynamic web pages so that the student can read the instruments and operate the devices according to each experiment. The same feature is also exploited in [15].

In [16] LabVIEW is used as a data visualization solution for Digital Signal Processor – 2 (DSP-2) systems. A LabVIEW toolkit was developed in order to serve on-the-fly data visualization and parameter tuning for the DSP-2 control systems. Functions for VI initialization, transmitting and receiving messages and other low-level routines were implemented inside. LabVIEW's built-in Web Publishing Tool is also used for publishing Virtual Instruments on the web.

2.2 Based on Simulation-based platforms (Matlab/Omnet/NS2)

NS or the network simulator is a discrete event network simulator [17]. It is popular in academia for its extensibility (due to its open source model) and plentiful online documentation. NS is popularly used in the simulation of routing and multicast protocols, among others, and is heavily used in ad-hoc research. NS Supports an array of popular network protocols, offering simulation results for wired and wireless networks alike. It can be also used as limited functionality network emulator. NS is licensed for use under version 2 of the GNU General Public License. Built in C++, NS provides a simulation interface through an object oriented dialect of Tcl, called OTcl. The user describes a network topology by writing OTcl scripts, and then the main ns program simulates that topology with specified parameters.

OMNeT++ is a discrete event simulation environment [18]. Its primary application area is the simulation of communication networks, but because of its generic and flexible architecture, is successfully used in other areas like the simulation of complex IT systems, queuing networks or hardware architectures as well. OMNeT++ provides a component architecture for models. Components (modules) are programmed in C++, then assembled into larger components and models using a high-level language (NED). Reusability of models comes for free. OMNeT++ has extensive GUI support, and due to its modular architecture, the simulation kernel (and models) can be embedded easily into your applications. Although OMNeT++ is not a network simulator itself, it is currently gaining widespread popularity as a network simulation platform in the scientific community as well as in industrial settings, and building up a large user community.

MATLAB is a numerical computing environment and programming language [19]. Created by The MathWorks. MATLAB allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages. Although it specializes in numerical computing, an optional toolbox interfaces with the Maple symbolic engine, allowing it to be part of a full computer algebra system.

Simulink, developed by The MathWorks, is a tool for modeling, simulating and analyzing multidomain dynamic systems [20]. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. It offers tight integration with the rest of the MATLAB environment and both drive MATLAB or be scripted from it. Simulink is widely used in control theory and digital signal processing for multidomain simulation and design.

Matlab and Simulink is used by many researches as a development tool for simulations and real applications and is generally preferred for several reasons. Along with the necessary toolboxes they constitute a reliable, well known platform with a lot of technical support, commonly used in teaching control courses. Also the time needed to obtain the prototype and in the development is less than the time needed with other tools and plat-forms. Finally it provides tools for remote execution of programs, and for real-time execution on a physical system, through a data system acquisition, using a specific control algorithm.

In [21] users can reserve processes in available time slots, design their control algorithms within the Matlab/Simulink software environment and connect to some real processes. The use of Simulink permits students to work with a user friendly software environment in most automatic control classes, allowing them to change on-the-fly control parameters and to visualize controlled system signals.

In [27] is presented a client/server architecture where the Automatic Control Telelab server is based on the Matlab/Simulink environment allowing the user to design her/his own controller. The user defined controller merges with a Simulink model representing a process. The Matlab Real-Time workshop routine converts the output model into a C source file, which is compiled to obtain executable code.

In [22] NS2 is used for analyzing TCP, UDP, Trinomial and TEAR protocols, a work that mainly focuses on the best design of Simple Network Transport Protocol, for programming a network of Robots remotely.

In [26] is presented an architecture that provides a client server distributed environment that facilitates effective sharing of resources. Three function groups are separated: control, execution, and storage of models and results, so that simulation hardware becomes accessible to a wider audience and hardware requirements are kept to a minimum. In the specific paper an OMNeT++ simulator is implemented that contains three software components each designed for a specified purpose.

Quite common is the simultaneous use of Matlab/Simulink and LabVIEW. In [23] a DSP-2 learning module based system is based on the particular pair. MATLAB and Simulink along with Real-Time Workshop are used for control algorithm development, simulation, offline analysis and rapid executable code generation, while the LABVIEW virtual instrument is automatically generated during the binary code generation process, from Simulink model, where the user front end of created VI depends on special DSP-2 blocks used in the Simulink mode. Using Remote Panels (LabVIEW add-on toolkit), generated VI's can be easily viewed and controlled over the Internet. LabVIEW VI's can be published on the Internet with no additional programming and can be remotely observed or controlled by using only the standard web browser.

2.3 Platform Indepedent

While mostly used, LabVIEW is not the only solution for instrument modeling. Several other applications have been introduced in order to provide the access to real instruments and real test benches. However the quick evolution of the computer and communication systems in the last few years has made them obsolete in the following key points: implementation cost, maintenance cost, security, flexibility, portability and communication burden. Java technology provides platform independence, high level of portability, flexibility and it has specifically been developed for the implementation of secure distributed systems. Java (RMI/CORBA) and the Java Native Interface (JNI) is usually preferred for a client-server solution that realizes remote experiments in an even more economical way. Java is a free platform while LabVIEW is not.

In [24] students with a Java Runtime Environment installed browser, connect to a Web Server and download the client software as Java Applets and start them. Due to the modular structure of the system, extensions with new features are easy to implement. The server hardware includes a video capture card and a sound card for video and audio transmission. The real-time controller of the experiment is implemented on a different computer hardware with real-time operating system.

In [25] JNI has been exploited in order to interface developed Java classes to the low level drivers of the remote labs instruments and boards. The server is equipped with the right boards in order to control instruments through an IEEE 488 interface.

3 Fields of applications

Remote experiments and their applications fall into the engineering domain so in order to distinguish them, more detailed engineering principles need to be identified.

3.1 Electrical and Electronics Engineering

In [13] the developed experiment is designed for the courses and lectures regarding electromagnetism and magnetic measurement topics.

[14] shows how experimental approaches to an electrical measurement subject can be approached in a modern, multimedia way. The remote test bench presented a characterization of a three-wire, three phase load as a significant example.

The use of an Internet-based laboratory facility for offering a digital electronics laboratory course along with an integrated evaluation process was presented in [15].

A remote didactic laboratory for electrical measurements of the Politecnico di Milano was presented in [25].

3.2 Control Laboratories

In [21], ARTIST laboratory was designed to provide an efficient remote and distributed control laboratory.

In [28], a Web-based learning framework, which addresses students and learners of control engineering, is presented.

Another laboratory for control experiments is described in [29]. This paper provides an approach for the development of remote labs using Java, LabVIEW and eMersion.

3.3 Mechanical Engineering

In [30] a thermal science laboratory is converted to an electronic equipped laboratory using LabVIEW

In [31] a comprehensive virtual instrument developed with LabVIEW was used for measuring thermal conductivity of a non-Newtonian fluid.

3.4 Physics or Chemical Engineering

Applying LabVIEW in chemistry and physics laboratories was reported in [32].

An extensive presentation of remote laboratories in physics is done in [33].

3.5 Network Security

In [33] a design and implementation of an Intrusion Detection System Virtual Laboratory (IDSVL) is made with Java in order to aid learners in understanding and mastering structure and theory of Intrusion Detection Systems.

4 Conclusions

The way information is transmitted over the world is constantly changing along with the means of transferring knowledge from the teacher to the learner. New methods and teaching means are continuously emerging in a technologically evolving environment. Computers tend to play a more significant role in the teaching process providing (arguably) quality services with considerable cost reductions, especially in the area of engineering education. From most of the referenced papers it is clearly shown that in engineering courses, the traditional laboratory classes can be replaced totally with simulated or remote laboratories.

The new simulated or remote laboratories are expected to equally provide quality experiences to students from increasingly diverse backgrounds. However, a good course material requires additional designing, teaching and organizational skills utilizing computer capabilities under the principles of engineering. The selection of the environment that meets the demands of each engineering course is a critical point to the development of the application, some aspects of which are presented in the first part of this paper. And even the use of the right environment and equipment does not guarantee a problem free system. The proper techniques and methodology should be applied in parallel.

The changing nature of engineering courses requires frequent alterations to the course material in order to keep the lessons up to date. This process requires a flexible software that is able to both adjust to the new technologies and provide the ability to renew its services.

Although the initial cost of setting up the laboratories is high, if the right technology is selected then improvements can be achieved with minimal cost. Partnership with other universities or organizations can reduce the costs in the case of initially expensive equipment or expansion of existing equipment due to additional needs.

References

[1] Magin D., Kanapathipillai S., 2000. "Engineering students' understanding of the role of experimentation" European J. Eng. Education 25, 4, 351-358.

[2] Farrington P.A., Messimer S.L., Schroer B.J. 1994. Simulation and undergraduate en-gineering education: The technology reinvestment project (TRP). In Proceedings of the 1994 Winter Simulation Conference. Lake Buena Vista, FL. J. D. Tew et al., Eds. 1387–1393.

[3] Colwell C., Scalon E., Cooper M., 2002 "Using remote laboratories to extend access to science and engineering. Comput. And Education 38, 1-3, 65-76.

[4] Shin D., Yoon E. S., Lee K. Y., Lee E. S. 2002. A web-based, interactive virtual laboratory system for unit operations and process systems engineering education: Issues, de-sign and implementation.Computers and Chemical Eng. 26, 2, 319–330.

[5] Grant, A. 1995. The effective use of laboratories in undergraduate courses.Int. J. Me-chanical Eng. Education 23, 2, 95–101

[6] Papathanassiou A., Oster J., Baier P. W., 1999. A novel simulation concept of reduced computational cost for TD-CDMA mobile radio systems with adaptive antennas. In Proceedings of the Vehicular Technology Conference (VTC 1999), Fall. Amsterdam, the Netherlands. 218–222.

[7] Sonnenwald D. H., Whitton M. C., Maglaughlin K. L., 2003. Evaluating a scientific collaboratory: Results of a controlled experiment. ACM Trans. Comput. Hum. Interact 10, 2, 150–176.

[8] Zimmerli S., Steinemann M. A., Braun T. 2003. Educational environments: Resource management portal for laboratories using real devices on the Internet. ACM SIGCOMM Comput. Commun. Review 53, 3, 145–151.

[9] Scanlon E., Colwell C., Cooper M., Paolo T. D, 2004. Remote experiments, reversioning and rethinking science learning. Comput. and Education 43, 1–2, 153–163.

[10] Nedic Z., Machotka J., Nafalski A. 2003. Remote laboratories versus virtual and real laboratories. In Proceedings of the 2003 33rd Annual Frontiers in Education Conference. Boulder, CO. T3E.1-T3E.6.

[11] T. Dehne, Putting Virtual Instruments to Work, Product Design and Development, Feature Report, No.360795A-01, National Instruments Corp. (1995).

[12] http://en.wikipedia.org/wiki/Labview, 12:55, 25 January 2008 Onorem

[13] Mladen Borsic et al., 2006 "Italian-Croatian Remote Laboratory Distributed on Geographical Network"

[14] Ferrero A., Salicone S, 2007 "Towards an hypertext of electric measurement: different approaches for an on-line, remote, didactic lab", IEEE Transactions on instrumentation and measurement, Vol. 56, No. 1

[15] Abul K. M. Azad, 2007, Delivering a Remote Laboratory Course within an Under-graduate Program, iJOE – Vol. 3, No. 4, 2007

[16] Darko Hercog, Suzana Uran, 2007, A DSP-Based Remote Control Laboratory, IEEE Transactions on industrial electronics, vol. 54, no. 6

[17] http://en.wikipedia.org/wiki/Ns2, 06:38, 27 January 2008 Md imranullah

[18] http://www.omnetpp.org/external/whatis.php

[19] http://en.wikipedia.org/wiki/Matlab, 20:51, 30 January 2008 85.64.15.17

[20] http://en.wikipedia.org/wiki/Simulink, 09:41, 19 January 2008 202.225.18.34

[21] Michele Basso, Gianni Bagni, 2004, ARTIST: A Real-Time Interactive Simulink-based Telelab, Proceedings of the 2004 IEEE, Conference on Computer Aided Control Systems Design, Taipei, Taiwan

[22] Raul Wirz, Raul Marin, Jose M. Claver, Josep Fernandez, Enric Cervera, 2007, Transport Protocols for Remote Programming of Network Robots within the context of Telelaboratories for Education: A comperative Analysis

[23] S. Uran, D. Hercog, K. Jezernik, 2006, Remote Lab Experiment RC Oscillator for Learning of Control, International Journal of Online Engineering (iJOE)

[24] C. Röhrig, and A. Jochheim, Java-based Framework for Remote Access to Laboratory Experiments, in Proc. IFAC/IEEE Symposium on Advances in Control Education, 2000, Gold Coast, Australia.

[25] Ferrero, A., Salicone, S., Bonora, C., and Parmigiani, M. 2003. ReMLab: A Java-Based remote, didactic

measurement laboratory. IEEE Trans. Instrumentation and Measurement 52, 3, 710-715.

[26] Erdei Mark, Soja Katalin, Wagner Ambrus, A networked remote simulation architecture and its remote omnet++ implementation, The 2nd International OMNeT++ Workshop, TU Berlin January 8-9, 2002

[27] Marco Casini et al., 2004 The Automatic Control Telelab: A web-based technology for distance learning, IEEE Control Systems Magazine

[28] C. Schmid, A. Ali, 2000, A Web-based system for control engineering education, American Control Conference 2000

[29] Sebastian Dormido et al., 2007, Using Web-based laboratories for control engineering education, International Conference on Engineering Education, ICEE 2007

[30] C. V. Knight and G. H. McDonald,1999, Modernization of a mechanical engineering laboratory using data acquisition with LabVIEW, ASEE,

[31] M. Kostic, 1999, Instrumentation with computerized data acquisition for an innovative thermal conductivity apparatus, ASEE,

[32] R. A. Young, 1999, LabVIEW: A Modern Data Acquisition System in an Introductory Mechanics Laboratory', ASEE

[33] U. Harms,2000, Virtual and Remote Labs in Physics Education, Proceedings of the Second European Conference on Physics Teaching in Engineering Education

ISSN 1613-5652
ISBN 978-3-940793-17-1