



Chapter 8

Easy Remote ISES – Development Environment Remote Experiments

MICHAL KRBEČEK¹, FRANTIŠEK SCHAUER^{1,2}
and FRANTIŠEK LUSTIG³

¹*Tomas Bata University in Zlín, Faculty of Applied Informatics, Department of Electronics and Measurements, CZ-760 05 Zlín, Czech Republic. E-mail: krbecek@fai.utb.cz.* ²*University of Trnava, Faculty of Education, Department of Physics, SK-918 43 Trnava, Slovak Republic. E-mail: fschauer@fai.utb.cz.* ³*Charles University, Faculty of Mathematics and Physics, CZ-12116 Prague -2, Czech Republic. E-mail: FL@plk.mff.cuni.cz*

This paper presents concerns the building of remote e-laboratories based on the physical software ISES (Internet School Experimental System), and presents the new environment – the EASY REMOTE ISES - for the plug and play compiling of the control program and communicating web page without the need for programming. The control system of ISES for remote experiments is an invariable finite-state machine for which the control file and corresponding web page are built for the particular experiment using a screen questionnaire similar to expert systems. The system responds to the wide shift of interest in remote experimentation especially in schools and universities, in order to help disseminate e-laboratories to a wider population.

INTRODUCTION AND STATE OF THE ART

In 2011, the Community Research and Development Information Service (CORDIS) of the European Commission issued a report on the 2011 Work Programmes of the Seventh Framework Programme (FP7). The report deals in part with the challenges facing Information and Communication Technologies (ICT) and with the “Target outcomes” in “Educational technologies for science, technology and maths”. The objective (Objective ICT-2011.8.1) on technology-enhanced learning is described, with “Target outcomes”

stated in part as: “(i) *Supporting Europeanwide federation and use of remote laboratories and virtual experimentations for learning and teaching purposes. The service shall enable online interactive experimentations by accessing and controlling real instruments, or using simulated solutions.* (ii) *Open interfacing components for easy plug-and-play of remote and virtual labs should be made available to stimulate the growth of the network of labs. Research shall include work on the user interfaces that mediate the complexities of creation and usability of* (iii) *experiments for specific pedagogical contexts in primary and secondary schools and higher education, including at university level.*” (See: ftp://ftp.cordis.europa.eu/pub/ftp7/docs/wp/cooperation/ict/c-wp-201101_en.pdf, page 91. Numbering of sentences added by authors)

Let us make a cross section of state of the art in the field of e-laboratories with a special focus on remote experimentation at schools in light of the assignment of the EU ICT-2011.8.1 project. The first point of the assignment (i), concerning the development of online interactive experimentation, is obvious and is reasonably in progress as documented in recent monographs from the field of remote experimentation [1-2].

A great deal of attention worldwide has been devoted to remote experimentation and especially remote e-laboratories, and these are under a strong current of evolution [1-2]. Technology enabled labs include different kinds of experiments, as we can distinguish among remote, hands-on and simulated laboratories [3]. Remote laboratory experiments are further distinguished by batched experiments, interactive experiments and sensor experiments [4]. Simultaneously, a large debate is still going on addressing the critical issue of whether remotely operated or simulation-based labs are as effective as the traditional hands-on lab format [5]. Meanwhile, online education has been strongly enhanced after the Bologna Declaration, which has contributed to shifting the learning focus from the institution to the student [6], with an emphasis on “learning by doing” using laboratories [7]. In this scenario laboratory activities are considered essential to increase the effectiveness of teaching and learning in scientific disciplines. On the other hand, laboratory management can be resource-intensive and expensive, since it requires qualified staff, continuous equipment maintenance and evolution, so that the number of laboratories is often limited. Also having a significant influence are the economic and logistics factors. For these reasons, the adoption of alternative access modes like remote laboratories are desirable to schools and universities [3].

Remote labs, in facts, can extend the capability of a conventional laboratory by increasing the number of times and places in which a student can perform experiments [8] while also extending the availability of the laboratory to several students. Moreover, they have the potential to provide affordable experimental data by sharing expensive laboratory equipment within a larger pool of users. The U.S. Accreditation Board for Engineering and Technology, now known as ABET, has taken on the task of assessing whether remote laboratories can truly accomplish the goals of educational laboratories and whether they – the remote laboratories – are adequate substitutions for traditional hands-on experience. For this purpose, a list of learning objectives was created describing the goals of the engineering laboratory [9].

Remotely controlled real labs are configured according to three different scenarios: Plug and play remote laboratory are used by students and researchers, according to a schedule supervised by the lab manager [10]. Remote collaborative laboratory where any allowed remote user can schedule a lab session, even in collaboration with other users. Different user types (researchers, university students, external users, etc) must observe

different management policies and scheduling rules, according to equipment features and lab manager requirements. Integration into LMS system like MOODLE [11] and Personal Learning Environments like Graaasp [12] is expected. In a blended remote laboratory, local and remote users can collaborate to use the lab equipment (e.g. a local tutor can coach a remote group of students about a new lab technique). The integration of remote labs into the university organization, so that they can become a tool to deliver laboratory services through the Web, is under way [13].

Gravier et al. [14] and Lowe et al. [15] propose five issues for the leverage of remote laboratories, and these are discussed in the following.

Collaboration. Remote laboratories should catch up with today’s learning theories of learning, namely constructivism, which emerged from cognitive science. Constructivism tries to make students learn from their own observations, using discussions with the teacher but also with their peers [16] and the learning outcomes of lab classes are strictly related to social interaction leading to Computer Supported Collaborative Learning (CSCL) [17]. The technologies used for building and collaboration of e-laboratories are depicted in Figure 2 [14].

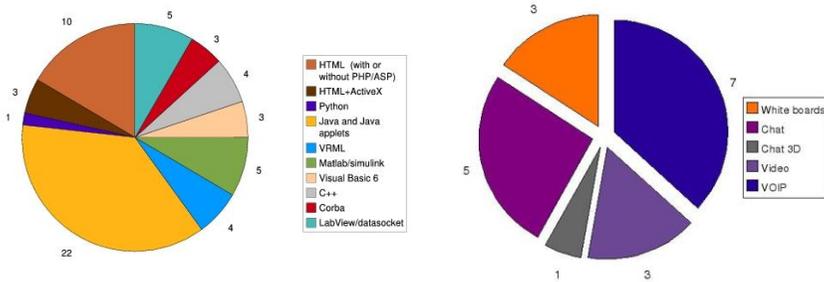


FIGURE 1 (LEFT) AND FIGURE 2 (RIGHT) [14]

FIGURE 1: TECHNOLOGIES USED FOR SERVER-CLIENT CONNECTION IN E-LABORATORIES

FIGURE 2: TECHNOLOGIES USED FOR COLLABORATION IN E-LABORATORIES

Convergence with Learning Management System. Distance learning is not a new matter of interest. Research is underway for a long time in order to deliver the best learning content exposition to the students and the best learning content reusability and production facilities to the teachers. The first issue leads to the creation of Learning Management Systems (LMS) and the second to the creation of Learning Content Management Systems (LCMS).

Evolution towards e-services. This calls for universities and research institutions to develop novel business and organizational models that extend their online offerings with remote laboratory activities for students and external users.

The second point of the EU ICT-2011.8.1 project assignment calls for: (ii) Open interfacing components for easy plug-and-play of remote and virtual labs should be made available to stimulate the growth of the network of labs. Research shall include work on the user interfaces that mediate the complexities of creation and usability of. In this respect Gravier et al. [14] have discussed recent progress made.

Reusability. The multitude of different technologies used implies that few efforts are made towards the reusability of existing remote laboratories (see Figure 1) [14]. A major issue is to propose not only solutions that decrease the time of integration, but also the amount of skills required in order to do it, starting from the approach of compiling the World Wide Web controlling page using the iconic environment similar to Easy Java Simulation (EJS) [18]. A second enhancement for plug and play devices will come from the Collaborative Remote Control Kit (CRC-Kit) [15].

Interoperability. In practice, remote laboratories can be composed of several devices that create an experimental virtual workbench when connected together. The interoperability can be reached using web services [19] and providing security, quality of service, and federated operation.

These above mentioned requirements set a list of technical and organizational challenges that should be specifically addressed. How to create and deliver easy plug and play HW and SW solutions for mediating the complexities of usage and creation especially of remote real and to create and deliver all the mature and excellent solutions to the wide population of interested and to be users of these techniques? The plug and play system for remote experiments at schools, both on HW side and SW side, with corresponding middleware, has been never realized as a complex problem till now.

Also the third point of the EU ICT-2011.8.1 project assignment (iii) *experiments for specific pedagogical contexts in primary and secondary schools and higher education, including at university level*, has not received corresponding attention till now. Besides the technical problems connected with remote experimentation it is necessary to solve the methodological problems in introducing the remote experimentation in the teaching and learning process. Real remote experiments bridge the gap till now existing in all forms of e-learning and a new strategy of education, based on the process, similar to the cognition of real world by the sciences, was introduced by Integrated e-Learning (INTe-L) [20].

The purpose of the present work is to help in ensuing situation and to report about a new environment – EASY REMOTE - ISES (ER-ISES) - for creating the control SW for remote e-laboratories based on physical software of Internet School Experimental System (ISES) by the plug and play approach using the screen questionnaire similar to the expert systems. The output is the control file and corresponding web page built automatically for the particular experiment. The new design environment and the paper itself are aimed at the wide shift of interested in the remote experimentation on Natural sciences, Technology and Mathematics in primary and secondary schools and higher education, including university level.

PLUG AND PLAY IN EXPERIMENTATION

Plug and Play in Informatics refers to an approach whereby devices work with a computer system as soon as they are connected. The user does not have to manually install drivers for the device or even tell the computer that a new device has been added. Instead the computer automatically recognizes the device, loads new drivers for the hardware if needed, and begins to work with the newly connected device [21]. This definition of plug and play comes from PC hardware and was enforced by the fact that the process of manually configuring devices, comprising the PC and its peripherals, could be quite difficult, and there was usually no forgiveness for technical inexperience. Incorrect settings could render the whole system or just the expansion devices completely or partially inoperable.

The Internet School Experimental System (ISES) - Plug and Play Hardware

ISES [22] is a software and hardware building set for plug and play. The ISES system was described in detail elsewhere [23] and here we give only a few relevant details.



FIGURE 3

PLUG AND PLAY PHYSICAL HW OF INTERNET SCHOOL EXPERIMENTAL SYSTEM: THE ADDA CARD AND SET OF ISES MODULES (LEFT) AND THE ISES PANEL WITH TWO ISES MODULES (RIGHT)

ISES is a complex tool for real time data acquisition, processing, displaying and control of experiments. ISES is an open system consisting of the ISES physical hardware and informatics hardware. The physical hardware is composed of the ISES panel and the set of modules (Figure 3a) and sensing ISES elements are easily interchangeable; their presence and adjusted range are automatically sensed by the computer, with the automatic calibration facility. The ISES panel (Figure 3b) enables 10 different channels (6 analog and 2 binary) and has capability to use 2 programmable outputs. The informatics hardware is composed of the interface card with A/DD/A converters and the informatics software is the controlling program (ISESWIN). The general scheme of ISES is depicted in Figure 4a.

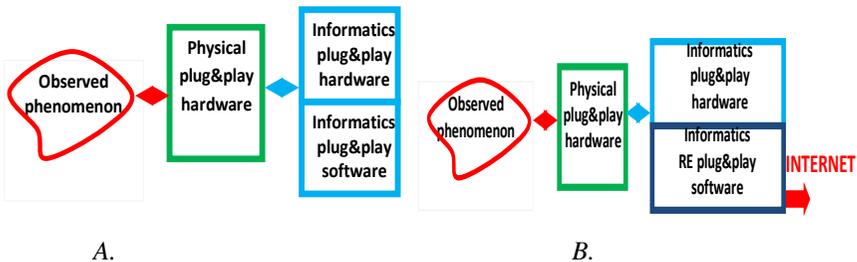


FIGURE 4

PLUG AND PLAY HARDWARE AND SOFTWARE SCHEMATIC REPRESENTATION ISES: HANDS ON EXPERIMENT (A) AND REMOTE EXPERIMENT (B)

Remote Experimentation with ISES

Today, there have been to our knowledge only two school systems with the software support for creating RE - LabVIEW and ISES. Let us discuss the present state of creating RE for schools by ISES in more detail. We started to elaborate this approach in 2000 [10]. Once we had the hands-on experiment, it was necessary to transform it into the RE by changing the controlling software, as depicted in Figure 4b, using the ISES WEB CONTROL kit [24]. This is the software that installs all the necessary components for RE as a finite-state machine (FSM) [25]. FSM is a mathematical model of computation used to design both computer programs and sequential logic circuits. It is conceived as an abstract machine that can be in one of a finite number of states. The machine is in only one state at a time. It can change from one state to another when initiated by a triggering event or condition, this is called a transition. A particular FSM is defined by a list of its states, and the triggering condition for each transition. The finite-state machine built on ISES and published [26] creates a controlling program for the specific RE functioning; it

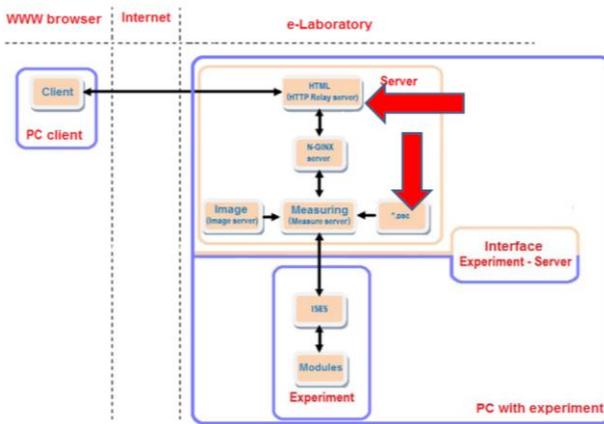


FIGURE 5

BLOCK DIAGRAM OF THE FINITE-STATE MACHINE (FSM) (TWO PRINCIPAL BLOCKS, THE PSC CONTROL FILE AND WEB PAGE MODULES ARE HIGHLIGHTED)

is depicted in Figure 5 with psc file and controlling web page (see arrows). The procedure of psc compiling was not straightforward, as it required knowledge of a special programming language, built for the purpose. All experiments built on ISES till 2012 (about 50 in number) have employed psc file programmed using this language transforming the flow chart diagram of the experiment to the controlling psc file [27]. The procedure was cumbersome and in effect was a hindrance for building remote experiments at schools, which now have a total of 450 computer laboratories with ISES.

Next we will describe new environment for plug and play controlling program compiling using the screen questionnaire similar to the expert systems, namely, EASY REMOTE ISES (ER ISES).

EASY REMOTE ISES

ER ISES is a graphical development environment which generates the control psc file for finite-state machine (FSM) as shown in Figure 5 and the web page code for RE controlling by a client (see also Figure 4b). Let us briefly introduce the development environment ER-ISES and how to work with it.

When we start the development environment ER ISES, the welcome menu with photos of available ISES versions appears showing the options (Figure 6). The welcome screen allows also the choice of language, to open the saved projects. For practical reasons the remote experiments are divided into three categories according to their complexity- **starting level** (set of precompiled typical RE), **basic level** (the most used RE with a variability and logics) and **advanced level** (very complicated RE). Let's describe the main parts of the program for the three RE complexity levels.



FIGURE 6
WELCOME PANEL OF EASY REMOTE ISES
GRAPHICAL DEVELOPMENT ENVIRONMENT

Starting Level

The simplest way of ISESRE building is to use the library of common experiments that the user can choose, automatically install, and start. The library is accessible by the button “Library of Experiments”, located on the main application panel (Figure 6). A window as shown in Figure 7 appears. In the upper left part of the window there is a list of experiments for the choice. When the experiment from the library is chosen, the description of the experiment appears in the right panel and beneath the list of hardware modules required. At the bottom left, there is a preview of the layout of the corresponding website (the shaded area). If you point the mouse to the edge of this screen (not accessible now) the arrows for the next screen appear and the photo of devices and modules are shown. In the last part is a text with description of the experiment. To end the compiling of the starting level remote experiment, it is then only necessary to press the “Finish” button. The program itself generates the necessary components of the experiment (psc file and web page code) and then returns to the start menu. At this point, the experiment is already operational, and we can start to use it. The whole procedure of RE compiling takes about 30 s.

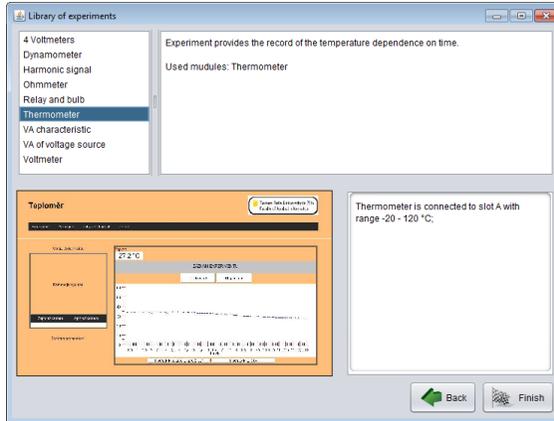


FIGURE 7
EXPERIMENT LIBRARY WINDOW OF ER ISES
FOR STARTING LEVEL OF RE OF COMPILING

Basic Level

In the basic level of RE, the ISES version has to be specified first. The choices are given in the welcome screen (Figure 6). Then the screen for the selection of measuring modules and their range adjustment occurs (Figure 8).



FIGURE 8
MODULE SELECTION WINDOW FOR BASIC LEVEL OF RE

The selection of modules is by the pull-down menu on the left side of the window. When a module is selected, its photo appears in the appropriate slot of the ISES panel. Its range is adjusted using the pull-down menu, available below the module.

If the experiment uses a relay board, one enables this option by pressing the “Relay board” button (Figure 8). Once pressed, a window appears with a relay board photo, by means of which you may connect and activate 2x8 relays. Then, the program asks for details of the individual relays settings. For example, we may choose manual activation with a button on a web page, or after meeting any comparative condition (e.g. comparing

an arbitrary reading of any quantity and any preselected value of it, or we may compare it with the instantaneous reading of a signal generated by ISES output).

Then the next screen for website design appears (Figure 9). On the left side of the screen is the name of the experiment and its IP address from which the experiment will be available globally. For local operation, press the button “Run experiment locally”. In this case, the experiment runs at the internal address 127.0.0.1 and it can be opened only from the computer on which it was installed. Next, we can add the logo that will be displayed on the web page and also the schematic arrangement or photo of the experiment. The other buttons are used to set the colors of the page header and footer.

The next important purpose of the screen in Figure 9 is the activation of selected

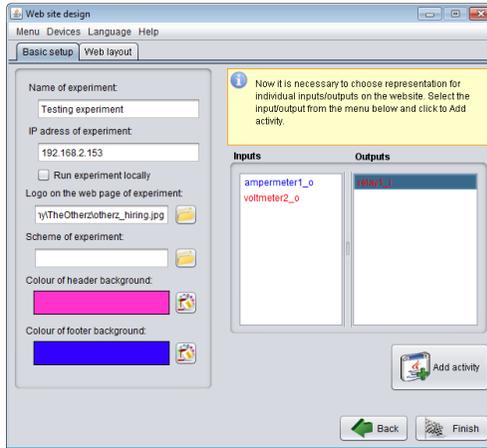


FIGURE 9
IP ADDRESS AND WEBSITE DESIGN WINDOW

modules. Module names are displayed in two columns (inputs and outputs). The red module displayed indicates the non-activated state, while the blue one shows the already correctly activated state of the module. To activate a module and insert the time representation of its readings (e.g. v-meter2_o) “Add activity” must be addressed and the catalogue of “activities” is shown (Figure10a). From the list we may choose the activity (“Time dependence (1variable)”). After selection of a desired activity click on “Add to page” button and the window with parameters for adjustment of the activity will be shown (Figure 10b). Most of the parameters are already filled, and must be confirmed only.

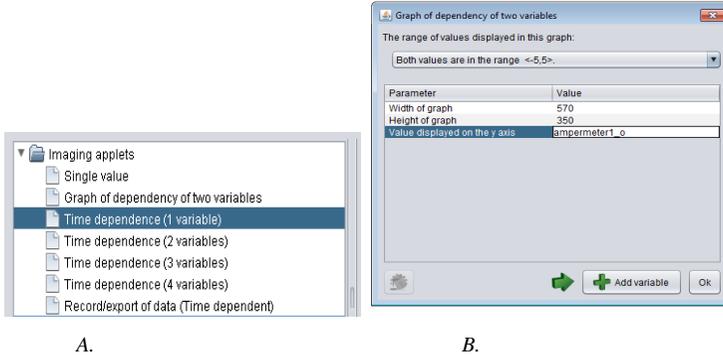


FIGURE 10

WINDOW FOR ACTIVITY CHOICE (A) AND WINDOW FOR ACTIVITY PARAMETERS ADJUSTMENT (B)

After activating all modules in the list of inputs and outputs, press the “Finish” button and the program goes back to the welcome menu. At this moment the RE is ready for use.

Advanced Level

The advanced level of RE compiling is based on the individual steps of the corresponding flowchart diagram and its transfer of the psc file by using the pre-prepared control blocks of the psc library. In this way it is already possible to assemble very complex control logic program for complex ISES RE. This method requires a certain amount of creativity and logical thinking concerning the sequence of actions of the experiment.

The advanced level of compiling the ISES RE starts on the module selection window (Figure 8) by the pressing “Advanced design” toggle button. Process of advanced design is similar to the basic design, where we have to select the modules and design the website. Only then the window for advanced level shows up (Figure 11). Its left part is a tree structure representing the control logic of the experiment. On the right side we find a list of blocks that can be inserted into a tree structure and a list of variables.

Compiling of control logic is performed by selecting the appropriate block from the menu and then clicking to the point of the tree, where the block is to be placed. Individual blocks have their own rules for inserting into the tree and may not be placed arbitrarily. Upon entering the advanced design, there is already in place the first block “State” in the tree. This block is the main block of the entire design representing certain step performed by the experiment. As an example we can mention the experiment: water level management, which controls the liquid level between two probes. In this experiment we distinguish states such as: filling, draining and waiting for switching of the probe. Activities of the individual steps are obvious from their names. For a new experiment we have to consider all actions present in the experiment and divide them into similar states before the start of controlling program compiling. The best way is to create a flowchart diagram of the experiment and to divide it into individual states before start. In practice, each state has three basic parts: entry, exit and step. These parts vary in time of execution

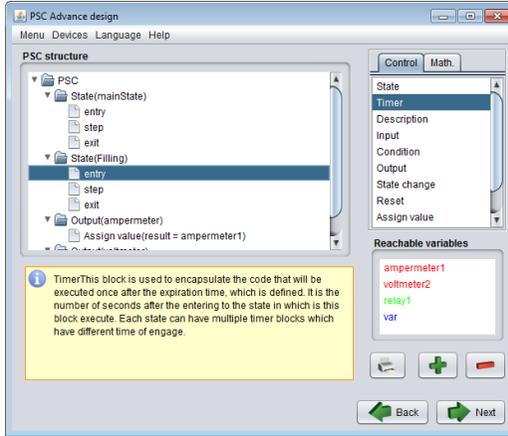


FIGURE 11
ADVANCED DESIGN WINDOW

of the code they represent. The part “entry” is always executed once, when the experiment enters the state, and similarly the “exit” part is executed on leaving the state. The middle part “step” is repeated until there is a transition to another state. The sequence of operations is shown in the following diagram (Figure12). After completion of the structure by the next button we can proceed to the website design.



FIGURE 12
SEQUENCE OF OPERATIONS IN ONE STATE

Other Functions of ER ISES

The development environment ER ISES provides storing of an unfinished draft of a RE. This makes the reediting of experiment very easy. This procedure may greatly shorten the design work in case of some errors. Saving of experiment (it is possible only as the last step of the webpage design) is by the “Save” button in the pop-up menu. Experiments are stored in a special file with extension .ers. Opening of saved experiment can be made from anywhere in the pop-up menu, or by the “Open” button directly on the welcome screen of the development environment. The opened experiment is shown in the initial step of the design. When one moves to the next steps, the data are already filled by the same values as in the moment of saving of the project.

APPLICATION OF ER-ISES ENVIRONMENT – AN EXAMPLE

In order to demonstrate the control file and the controlling client’s web page design in the development environment ERISES, we present the example of the creation step by step. We choose for this purpose the well known interactive experiment of Electromagnetic

induction in the coil rotating in a magnetic field. We require the choice of the frequency of rotation of the coil, graphical time representation of its instantaneous voltage, data transfer to the client's computer, and the live view of the experiment. Let us suppose we have the hands-on ISES experiment ready where for driving the coil rotation we use a dc motor (0-5V) and have a web camera for our disposal (Figure13).

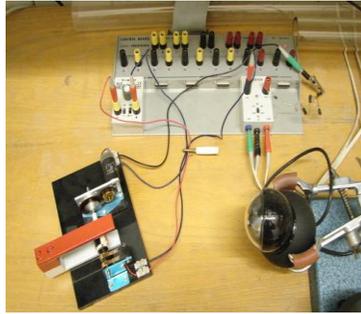
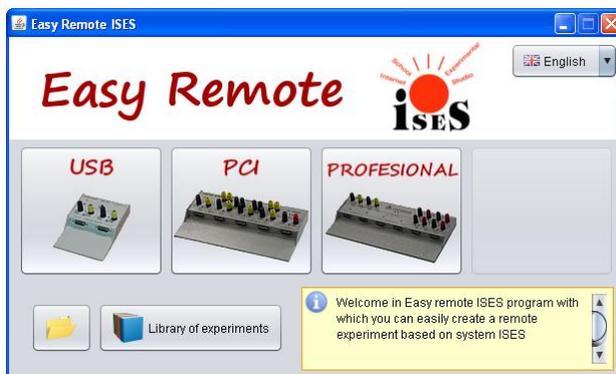


FIGURE 13

EXPERIMENT ARRANGEMENT OF THE ISES REMOTE EXPERIMENT
ELECTROMAGNETIC INDUCTION (LEFT: COIL WITH A DRIVING MOTOR AND
THE PERMANENTMAGNET; TOP: ISES PANEL WITH V-METER ISES, AND
OUTPUT ISES MODULE FOR MOTOR DRIVING; RIGHT: CAMERA

Let us choose the basic level design of the controlling program, with the following steps:

1. Start the development environment ER ISES and select the ISES version. In our case, we chose PCI ISES - Demo Figure 1.



DEMO FIGURE 1

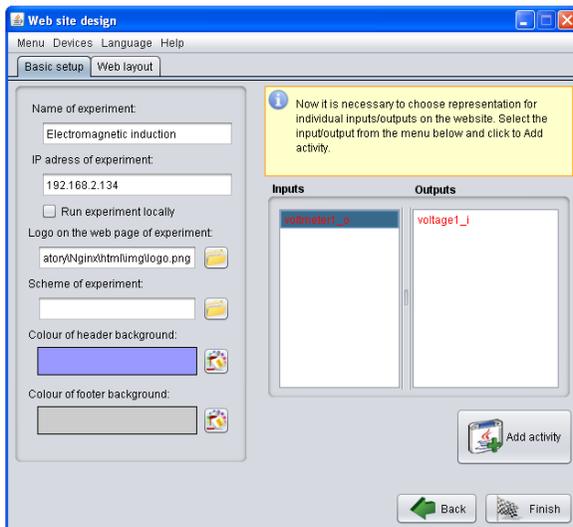
2. Demo Figure 2 appears. Now we need select the modules for the signal measurement of the coil voltage - ISES V-meter module in the input A. Choose



DEMO FIGURE 2

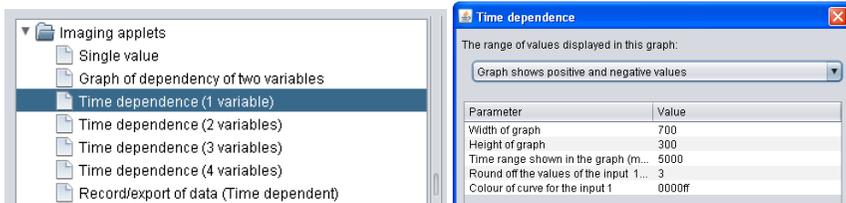
the V-meter module from the drop-down menu and adjust its range. For driving the motor we will select the output E as depicted in Demo Figure 2. We will supply the driving motor directly from the ISES board; therefore, select “Voltage” as the output module. Depending on the motor, we may increase the output current by using the booster ISES module. After addressing “Next” button the program asks for details of the setting of the output voltage.

3. Now it is necessary to perform the basic setup of the website (Demo Figure 3). We may here enter a name for the experiment and the IP address on which the experiment is running, logo displayed on the page or change the color of the page background.



DEMO FIGURE 3

In the list of inputs we have the un-activated (red color) voltmeter module connected to position A. By activating “Add activity” we open the catalogue of activities (Demo Figure 4A), where we activate the module and select the representation of the voltmeter readings on the web page. Let us select “Time dependence (1variable)”. Then we are asked for the adjustment of graph’s parameters (Demo Figure 4B) and displaying of negative voltage in the graph. Then the V-meter module is activated (blue color for V-meter module occurs).



DEMO FIGURE 4A (LEFT) DEMO FIGURE 4B (RIGHT)

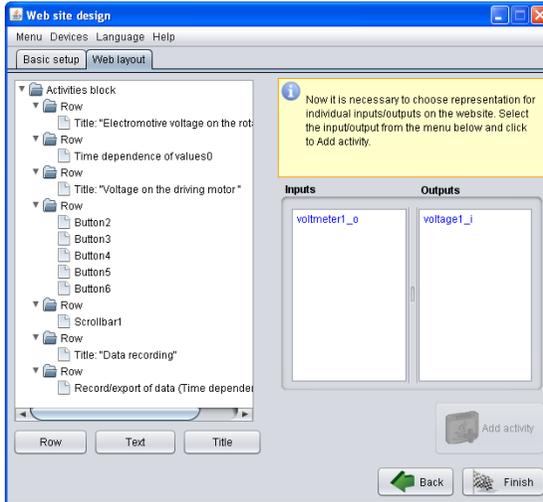
- Now we will activate the voltage for the driving motor. In the list of outputs (Demo Figure 3) we have the reading "voltage1_i" (in red color). We want to adjust this voltage in two ways. The first will be by using the activity "Scrollbar", which allows a continuous adjustment of the driving voltage. Let's insert it in the similar way as in the previous case using the offer "Scrollbar" in Demo 4A. There are no parameters in "Scrollbar" activity. The other way of the driving voltage adjustment is by the five fixed selected voltages (0, 2, 3, 4 and 5 V). For this purpose we insert the activity "button" (selected from the list of activities in Demo Figure 4A). For the settings of activity, select the "Setting the output voltage", enter a description of the button ("Stop") and voltage value 0V ("0") (Demo Figure 5). This button will turn off the driving motor. In the same way we insert four more buttons which will adjust fixed voltages 2, 3, 4 and 5 V for the driving motor.



DEMO FIGURE 5

- The last activity we add is the capture and storing of the measured data on the client's computer. This is done by the inserting the activity “Record/Export of data” (Demo Figure 4A).

- Let us modify the layout of activities on the web page. For this is the "Web layout" tab (see Demo Figure 3). In the tree structure, we can see all added activities (Demo Figure 6). We can use the buttons to add titles, labels and rows. By moving the applets in the tree and adding them to the rows, we can adjust the layout on the website.

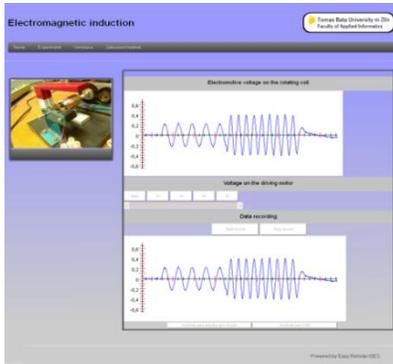


DEMO FIGURE 6

- Now we can finish the compiling of the remote experiment, but for safety reasons it is recommended to save it first. This is done by The "Save" item in the main upper Menu. We select the name and path of the saved file and confirm. Then we can finish the experiment by using the "Finish" button. Now the experiment is already running and we can access it via the given IP address. In the Appendix we show the printout of the generated psc file and similarly the web page is generated.
- The final web page layout and the sample of the exported data by RE Electromagnetic induction, is in Demo Figure 7.

FUTURE DEVELOPMENT AND CONCLUSIONS

The graphical development environment ER ISES is at present fully functional and is suitable for the compiling without any programming of both simple remote experiments and complex solutions requiring advanced control logic as well. At present development work continues in the direction of the iconic choice of steps in control program and web page compiling. Control logic will be created by the choice of graphic blocks instead of the existing activity solutions. Another goal is to operate our system of programs design online in a form of e-shop for both HW and SW choice. The Library of experiments,



No	Time t (s)	Voltage U (V)
1	0.009	-0.4019
2	0.018	-0.4726
3	0.03	-0.538
4	0.038	-0.5743
5	0.047	-0.5688
6	0.059	-0.5198
7	0.068	-0.4436
8	0.079	-0.313
9	0.088	-0.2273
10	0.097	-0.1098
11	0.109	0.0445
12	0.118	0.1624
13	0.129	0.2985
14	0.138	0.3947
15	0.15	0.4836
16	0.159	0.5235
17	0.167	0.5688
18	0.179	0.5579
19	0.188	0.4926

DEMO FIGURE 7

which is currently stored on the local disk, will in the future run on the web server of the data center, possibly in a cloud virtualized regime. Library will not only allow users to download experiments, but also upload created experiment back to the storage server. This will form online library of experiments that will provide a steady flow of new experiments formed by the users themselves.

The conclusions of this paper may be summarized as follows:

- We have developed and described the graphical development environment for remote experiments EASY REMOTE ISES that enables the design of the controlling software on the basis of server – client communication.
- The system is designed for three types of complexity of remote experiments by the plug and play approach without any programming, enabling those interested in remote experimentation especially at schools and universities to master this new and progressive technique for teaching.

ACKNOWLEDGMENT

This work is published thanks to the support by the Slovak grant agency KEGA No 011TTU-4/2012 and APVV project No. APVV 0096-11.

REFERENCES

1. J. Garcia-Zubia, and G. Alves, eds., "Using Remote Labs in Education," Ed. University of Deusto, Miller, F. P., 2009.
http://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CDsQFjAB&url=http%3A%2F%2Fwww.weblab.deusto.es%2Fweb%2Fweblab.content%2Fusing_remote_labs_in_education.pdf&ei=zCBXUbljiePhBMrgIgL&usg=AFQjCNGI6EDF2FUDieIDBnc5UiAqJkEl1g&sig2=y9mPmF6vtHJClmLdbWYMew&bvm=bv.44442042,d.bGE
2. K. Azad, M. Auer, J. Harward, "Internet Accessible Remote Laboratories: Scalable E-Learning Tools for Engineering and Science Disciplines," Ed. IGI Global. USA, 2012. DOI: 10.4018/978-1-61350-186-3, ISBN: 9781613501863. <http://www.igi-global.com/book/internet-accessible-remote-laboratories/52730>

3. J. Ma, and J. V. Nickerson, "Hands-On, Simulated, and Remote Laboratories: A Comparative Literature Review," *ACM Computing Surveys*, Vol. 38, 2006.
4. J. Garcia-Zubia, P. Orduna, D. Lopez-de-Ipina, G.. Alves, "Addressing Software Impact in the Design of Remote Laboratories," *IEEE Transactions on: Industrial Electronics*, Volume 56, Issue 12, pp. 4757-4767, 2009.
5. J. Corter, J. V. Nickerson, S. K. Esche, C. Chassapis, Im Seongah, and J. Ma, "Constructing Reality: A Study of Remote, Hands-On, and Simulated Laboratories," *ACM Transactions on Computer-Human Interaction*, Vol. 14, No. 2, Article 7, 2007.
6. T. Restivo, F. G. Almeida, M. F. Chouzal, J. G. Mendes, A. M. Lopes, J. P. Carneiro, and H. S.Lopes, "On the Way for a Better Methodology in Teaching/Learning Instrumentation for Measurements at Mechanical Engineering Compulsory Syllabus", in *Proc. WCETE*, , pp. 92–96, March 2004.
7. Z. Doulgeri, T. Matiakis, "A Web Telerobotic System to Teach Industrial Robot Path Planning and Control," *IEEE Trans. Educ.*, Vol. 49, No. 2, pp. 263–270, 2006.
8. G. Canfora, P. Daponte, and S. Rapuano, "Remotely Accessible Laboratory for Electronic Measurement Teaching," *Comput. Standards and Interfaces* 26, 6, 489–499, 2004.
9. A. Rosa, "The Challenge of Instructional Laboratories in Distance Education," Paper presented at the *2003 ABET Annual Meeting*, Baltimore, MD, ABET, 2003 October.
10. F. Lustig, F. Schauer and M. Ozvoldova, "Plug and Play System for Hands-On and Remote Laboratories," in *Proc. MPTL16-HSCI 2011*, Ljubljana, 15-17 September, 2011.
11. M. Bochicchio, A. Longo, "Extending LMS with Collaborative Remote Lab Features," *ICALT*, 310-314, 2010.
12. D. Gillet, E. Law, and A. Chatterjee, "Personal Learning Environments in a Global Higher Engineering Education," Web 2.0 Realm, *1st IEEE Engineering Education Conference (EDUCON)*, Madrid, Spain, April 14-16, 2010.
13. E. G. Guimarães, E. Cardozo, D. H. Moraes, and P.R. Coelho, "Design and Implementation Issues for Modern Remote Laboratories," *IEEE Transactions on Learning Technologies*, Vol. 4, No. 2, pp. 149-161, 2011.
14. C. Gravier, J. Fayolle, B. Bayard, M. Atesand, J. Lardon, "State of the Art about Remote Laboratories Paradigms - Foundations of Ongoing Mutations," *International Journal of Online Engineering*, Vol. 4, no. 1, pp. 19-25, 2008. <http://online-journals.org/index.php/i-joe/issue/view/42>
15. D. Lowe, C. Berry, S. Murray and E. Lindsay: "Adapting a Remote Laboratory Architecture to Support Collaboration and Supervision," *International Journal of Online Engineering*, Vol. 5, Special Issue REV 2009, pp. 51-56, 2009, doi:10.3991/ijoe.v5s1.932 , <http://www.online-journals.org/index.php/i-joe/issue/view/71>
16. D. Jonassenand, and S. Wang, "Acquiring Structural Knowledge from Semantically Structured Hypertext," *Journal of Computer-Based-Instruction*, 20: 1–8 (1993).
17. T. Koschmann, "Dewey's Contribution to the Foundations of CSCL Research," in G. Stahl (ed.), *Computer Support for Collaborative Learning: Foundations of a CSCL Community, Proceedings of CSCL 2002*, pp. 17–22, 2002.
18. F. Esquembre, and J. M. Zamarro, "EJS: An Authoring Tool to Develop Java Applications: Computers and Education," *Towards an Interconnected Society*, pp. 143-148, 2001.

19. Y. Yan, Y. Liang, X. Du, H. Saliyah-Hassane, and A. Ghorbani, "Putting Labs Online with Web Services," *IT Professional*, 8(2): 27–34, 2006.
20. F. Schauer, M. Ožvoldová and F. Lustig, "Integrated e-Learning – New Strategy of Cognition of Real World," in *Teaching Physics, Innovation 2009, World Innovations in Engineering Education and Research*, iNEER Spec. Vol., W. Aung et al. (eds.), 2009, USA, pp. 119-136.
21. Wikipedia, the free encyclopedia, "Plug and Play," http://en.wikipedia.org/wiki/Plug_and_play, read 15.10.2012.
22. F. Schauer, I. Kuřitka and F. Lustig, "Creative Laboratory Experiments for Basic Physics Using Computer Data Collection and Evaluation Exemplified on the Intelligent School Experimental System (ISES)," in *Innovations 2006, World Innovations in Engineering Education and Research*, iNEER Special Volume, W. Aung et al. (eds.), pp. 305-312, USA, 2006.
23. F. Schauer, F. Lustig and M. Ozvoldova, "ISES – Internet School Experimental System for Computer-Based Laboratories in Physics," *Innovations 2009, World Innovations in Engineering Education and Research*, iNEER Special Volume (USA), chapter 10, pp.109-118, 2009.
24. ISES Website, "E-Laboratory Project," www.ises.info.
25. Wikipedia, the free encyclopedia, "Finite-State Machine," http://en.wikipedia.org/wiki/Finite-state_machine
26. F. Schauer, F. Lustig, J. Dvorak and M. Ozvoldova, "An Easy-to-Build Remote Laboratory with Data Transfer Using the Internet School Experimental System," *Eur. J. Phys.* 29 (2008), 753-765.
27. Website Listing, <http://www.eEdu.eu> (in Czech).

Michal Krbeček received the MS degree in Informatics from Tomas Bata University in Zlin in 2011. Now he is a PhD student at the Department of Mathematics, Faculty of Applied Informatics, Tomas Bata University in Zlin, Czech Republic. He is active in software programming for remote experimentation.

František Schauer received the M.S. degree in Electronics from the Brno University of Technology in 1963 and the Ph.D. degree in Solid State Physics from Prague University of Technology in 1978. In 1982 he was appointed Associate Professor and in 1988 Professor in Condensed Matter Physics at the Technical Academy in Brno. In 1993-2002, he was with the Faculty of Chemistry, Brno University of Technology and, since then, he has been with the Faculty of Technology and Faculty of Informatics, Tomas Bata University in Zlin, Czech Republic. His main activities are molecular electronics, computer assisted experiments, and e-learning in physics teaching.

František Lustig received his M.S. degree in Didactics of Physics from the Charles University in Prague, Czech Republic, in 1976. He received his Ph.D. degree in Plasma Physics from Charles University in 1986. In 2005 he was appointed Associate Professor in Didactics of Physics. He is the author of ISES (Internet School Experimental System) and iSES (internet School Experimental

Studio). His main activities are computer-aided experiments, remote laboratories and videoconferences from an experimental laboratory.

APPENDIX

PSC Control Code Generated by EASY REMOTE ISES Environment

```
# Generated by EASY REMOTE ISES. Developed by Ing. Michal Krbecek et al.
# Version.
version 3.2
# Script name.
name control
step_frequency 111
# HW plugin
usingcard_lib "plugins\PCI1202CardPlugin.ldap" card_config_1
# Pins import
import voltmeter1 card_lib "Read A/D 1" remap <0;4095> to <0;10>
import voltage1 card_lib "Write D/A 1"
# Users variables
variable avg_voltage1 = 0.0
variable sum_voltage1 = 0.0
variable voltmeter1counter = 0.0
statemainState{
entry {
voltage1 = 2047.5;
}
step {
sum_voltage1 = sum_voltage1 + voltmeter1;
voltmeter1counter = voltmeter1counter + 1;
if (voltmeter1counter >= 3) {
avg_voltage1 = sum_voltage1 / voltmeter1counter;
sum_voltage1 = 0;
voltmeter1counter = 0;
}
}
pin_read voltmeter1_o {
result = avg_voltage1;
}
pin_write voltage1_i {
voltage1= new_value;
}
on_user_change{
voltmeter1counter = 0;
avg_voltage1 = 0;
sum_voltage1 = 0;
voltage1 = 0;
}
}
```