

Fields of Applications for Hybrid Online Labs

<http://dx.doi.org/10.3991/ijoe.v9iS3.2542>

K. Henke, St. Ostendorff, H.-D. Wuttke, T. Vietzke and Ch. Lutze
 Ilmenau University of Technology, Ilmenau, Germany

Abstract—Based on a grid concept of an interactive hybrid online laboratory we will describe different fields of applications in different learning scenarios. The infrastructure is based on a universal grid concept which guarantees a reliable, flexible as well as robust usage of this online lab. By using the online lab, students are able to design control algorithms with different specification techniques to control electro-mechanical models in the online lab. Additionally, the reconfigurable rapid prototyping platform of the REAL system can be used to test all the taught topics of a given lectures in the field of digital system design. Finally, a special demonstration platform (a ball in a labyrinth on a balance plate) can be used to give the students a better feeling about the possibilities and limitations of remote control and observation via Internet and to evaluate these technologies critically. The implemented online lab infrastructure is based on the iLab architecture of the MIT, which allows to interconnect online labs and to exchange remote lab experiments among different universities worldwide.

Index Terms—control engineering education, laboratories, Web-based education, virtual and remote labs, Web-based design tools, distance learning.

I. INTRODUCTION

Our Integrated Communication Systems Group at the Ilmenau University of Technology has many years of experience in integrated hard- and software systems and over 10 years of experience in dealing with Internet-supported teaching in the field of digital system design ([1] and [2]). We have developed a new teaching concept, called “Living Pictures” [3] that we use in several phases of the learning process. Living Pictures are highly interactive Java applets that can be used for demonstrations as well as for experimental purposes, and also serve as tools in certain steps of the design process of digital systems. To complete the learning outcomes by own experiences, the students have to pass hands on examination in a lab. A task during this examination is to design an algorithm for a control system that controls one of various physical systems, for instance an electro-mechanical model of an elevator or a production cell.

For all students, hands-on experiences are important to deepen their knowledge about topics they learned during lectures (see Figure 1). At our university we offer an online laboratory, which gives the students the possibility to work on real physical systems without the need to stand in line at a lab or the need to take care of opening hours.

With our hybrid¹ online lab we want to offer the students a working environment that is as close as possible to a real world laboratory. Under real laboratory conditions disturbances can appear and lead to failures of the control algorithm that cannot be detected under virtual lab conditions.

experiment objects		user location	
		local	remote
physical system	real	local lab	remote lab
	virtual	hybrid lab	hybrid online lab
simulation model of the physical system		local simulation	remote simulation (virtual lab)
		hands-on lab	online lab

Figure 1. Classification of lab experiments

Therefore, it is important to include such real disruptive factors for a closer relation to practical conditions. Furthermore, the online lab should offer the students stimulus with regards to the design of safety critical control systems. By the conception of our lab (described in the next section), the virtual worlds were embedded within a real laboratory.

In principle, we would like to concentrate on giving students the chance to check their prepared control algorithms in real environmental conditions, which they can interactively influence themselves, and correct or modify the received results

- via Web-based simulation,
- via Web-based remote control of the existing physical system (that means before the tutorial course).

Currently, there are no common standards for the architecture of remote labs. This is the reason why universities develop their own remote lab solutions – suitable for their specific requirements, for example [4], [5], [6] and [7]. This handicaps a networking of different online labs among each other and also a usage by different institutions. Missing uniform interfaces prevent the programming of universal plug-ins and other software modules.

¹ Hybrid online labs provide both remote experiments on real electro-mechanical models (physical systems) in the remote lab as well as simulation models of these physical systems in virtual labs.

II. ARCHITECTURE OF THE *REAL* SYSTEM

In the following, we will describe a hybrid interactive online lab, supporting all the design steps for complex control tasks to control various electro-mechanical models – the *REAL* system. Goal of the *REAL* (**R**emote and **A**pplications **L**aboratory) system is to show new ways and chances of remote controlling and remote observation of real processes (e.g., in the fields of control engineering, robotics, tele-control engineering), dealing with the integrated and interactive usage of modern Internet and intranet technologies, like HTML5, jQuery, JavaScript etc. It was developed at the Department of Integrated Communication Systems at the Ilmenau University of Technology [8] – see Figure 2.

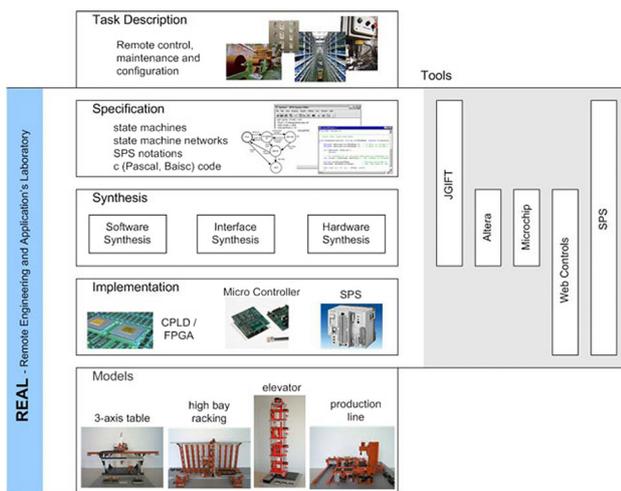


Figure 2. Overview of the *REAL* system

The implemented *REAL* infrastructure is based on the iLab Shared Architecture of the MIT (see Figure 3) which meanwhile is established as a standardized implementation for online laboratories and will be implemented in more and more locations all over the world. Furthermore, it allows to interconnect online labs and to exchange remote lab experiments among different universities worldwide.

As mentioned in many papers (e.g., [9], [10] and [11]), interactive online labs can open opportunities which allow an experimental approach for a wider audience and are also independent of opening hours of the laboratory rooms and their staff.

Interactive labs will be used, when:

- an experiment requires real-time processing or
- the user wants to observe the whole physical system (the electro-mechanical models) during the experiment or
- some parameters (e.g., input variables to a control system) need to be changed interactively during the experiment.

They offer various features like visualization and animation, which allows to observe and to test all the properties of the design. In connection with formal design techniques, simulation and prototyping are used to establish a foundation for the development of a reliable system design. To check the functionality of the whole design, some special simulation and validation features are included as integral part of the *REAL* system. This offers various possibilities for the execution of simulations, such as:

- usage of simulation models of the physical system for visual prototyping,
- step by step and parallel execution of these prototypes,
- visualization of the simulation process with the tools also used for specification,
- features for test pattern generation and
- code generation for hardware and software synthesis.

REAL offers a Web-based environment supporting the above mentioned features to generate and execute a design by using simulation models. An example will be given in Section III.A.

At any time the students have the chance to adjust their algorithms in case of faults. Therefore, they are able to achieve a fault free solution (a validated control algorithm) step by step. For more details see the previous publications [2], [12] and [13].

Our online lab is used for teaching practical lessons as well as giving hands-on experiences for the development of embedded electronics. This is not done using on-site lessons, but remote via the Internet. This has the advantage that courses can be offered internationally world-wide and gives students from different countries, speaking different languages, the same access and equal possibilities in the lab.



Figure 3. iLab Shared Architecture of the MIT [4]

Additionally, even for local students, the lab offers extended opening hours (twenty-four-seven) when compared to a regular lab. Besides the advantages for students, this also reduces the costs for academic teaching and improves the quality by offering more practical training possibilities.

One implementation challenge is to protect the physical systems in the lab against wrong control algorithms of students without defining too many design constraints. Students should be free in their decisions and develop own creative solutions. They can implement their own design strategies, therefore a reference design and a method to check the students' design against this reference is needed [14] to protect the physical system in the lab (see Figure 4).

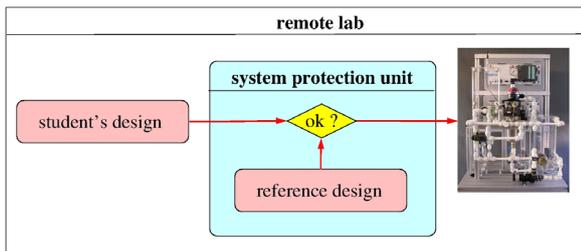


Figure 4. Observation of the student's design by the physical system protection unit

The reference design should be independent of the used control unit and the development tools. This is done by the protection unit of the physical system.

Figure 5 illustrates the grid architecture of the *REAL* system. The server side infrastructure (remote lab) consists of three parts:

- an internal serial *remote lab bus* to interconnect all parts of the remote lab,
- a *bus protection unit* to interface the control units to the remote lab bus and to protect the bus from blockage, misuse and damage as well as
- a *physical system protection unit*, which protects the physical systems (the electro-mechanical models in the remote lab) against deliberate damage or accidentally wrong control commands and which offers different access and control mechanisms.

For a more detailed description of this grid concept as well as of the main components see papers [15, 16].

The interconnection between the Web-control units and the selected physical systems during a remote lab work session (experiment) as well as the webcam handling is done by the lab server as part of the iLab architecture (see Figure 3).

The iLab Service Broker is mainly responsible for the user management and time scheduling. All experiments are made available by this service and integrated into the iLab Cloud.

During a running experiment, the client application will interact directly with the online lab infrastructure as depicted in Figure 5 without any more access to the Service Broker. Details about the iLab architecture are beyond the scope of this paper. Please refer to [17] for further information.

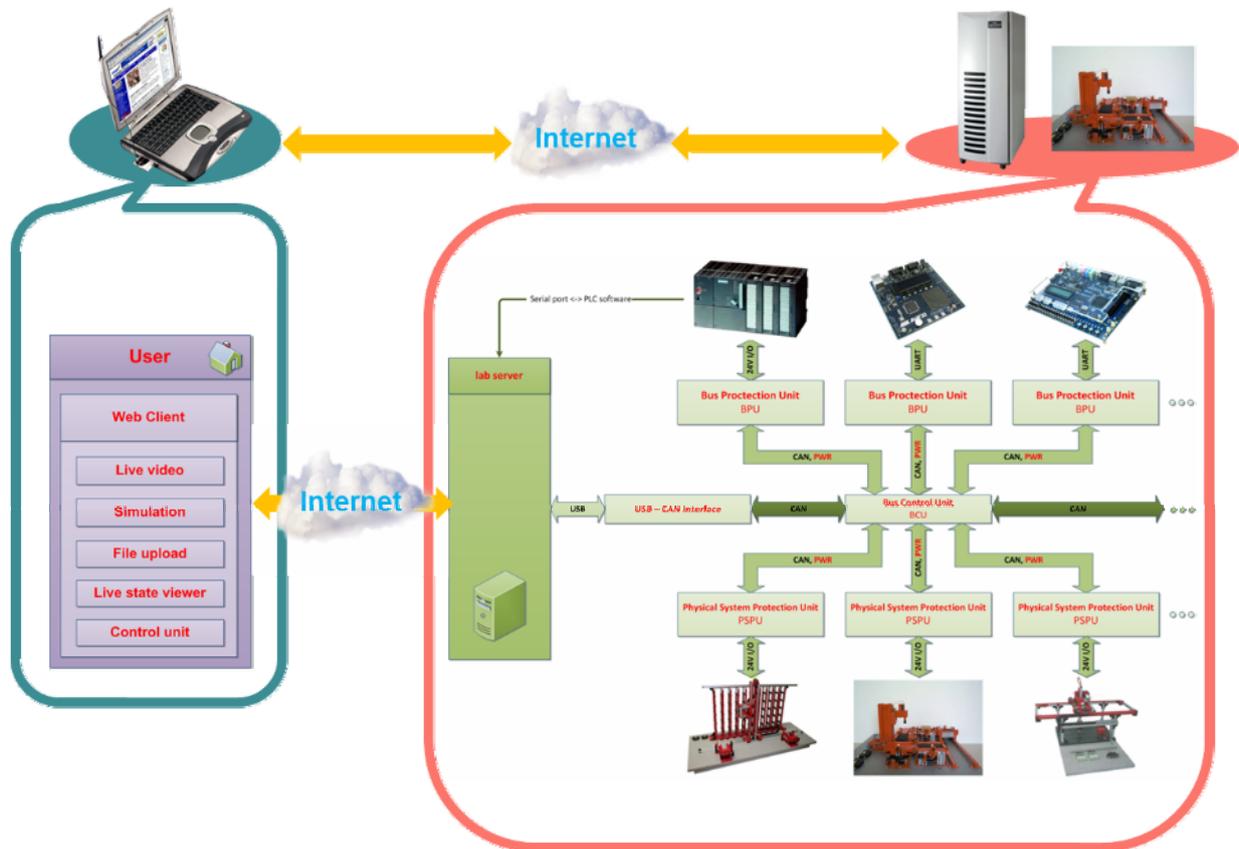


Figure 5. Grid architecture of the *REAL* system (server side) with Web-client

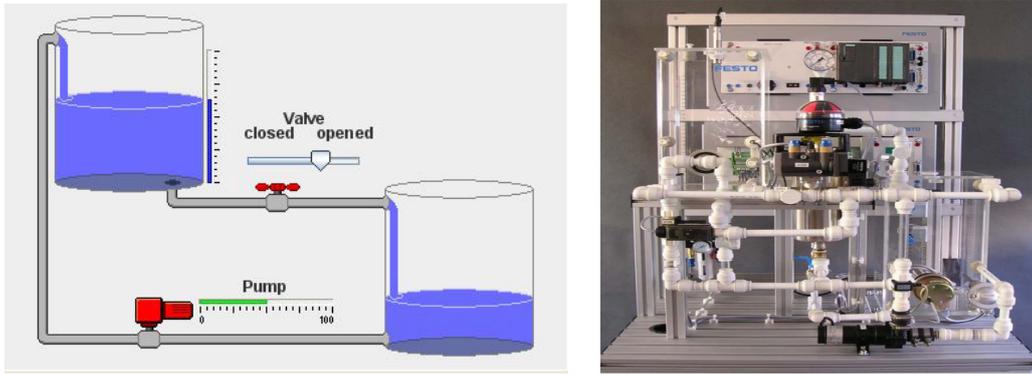


Figure 6. Simulation model and physical system of a water level control

III. FIELDS OF APPLICATIONS

Based on this flexible online lab structure we offer different operation modes to test the developed control task. Simulation and visual prototyping help to find functional errors. Before starting practical work on real systems, simulations and animations in “virtual worlds” are often used to verify the developed solutions. The behavior of the physical system that should be controlled, as well as its environment, will be emulated as a simulation model. The student can influence this “virtual world” and analyze the caused reaction of his control algorithm. Figure 6 shows an example of such a simulation model.

These steps have to be executed until no more errors are detected. But there is an essential disadvantage in this method. Real disruptive factors (e.g., failure of single components, mechanical problems or process variations) cannot be recognized by the underlying virtual environmental model.

Generally, only a simulation of predetermined malfunctions is possible. After some time, all these effects are well known in the student’s community. Unconsidered

sources of errors lead to undetected failures of the control because the corresponding environmental situation was not simulated before [13]. That is why a fault free design algorithm finally should be tested on real physical systems (e.g., the water level control, shown in Figure 6) in the online laboratory as well. In the following we will describe possible operation modes of the *REAL* system based on the schematic view in Figure 5:

- Stand-alone Mode (visual prototyping)
- Remote Control Mode (via Web-client)
- Remote Control Mode (via control unit)
- Virtual Control Mode (visual prototyping)
- Virtual Control Mode (test mode)
- Local Control Mode (via control unit)
- Local Control Mode (manually)
- Rapid Prototyping Mode
- Visitor Mode

The complete schematic view (client and server side) of the online lab architecture is shown in Figure 7.

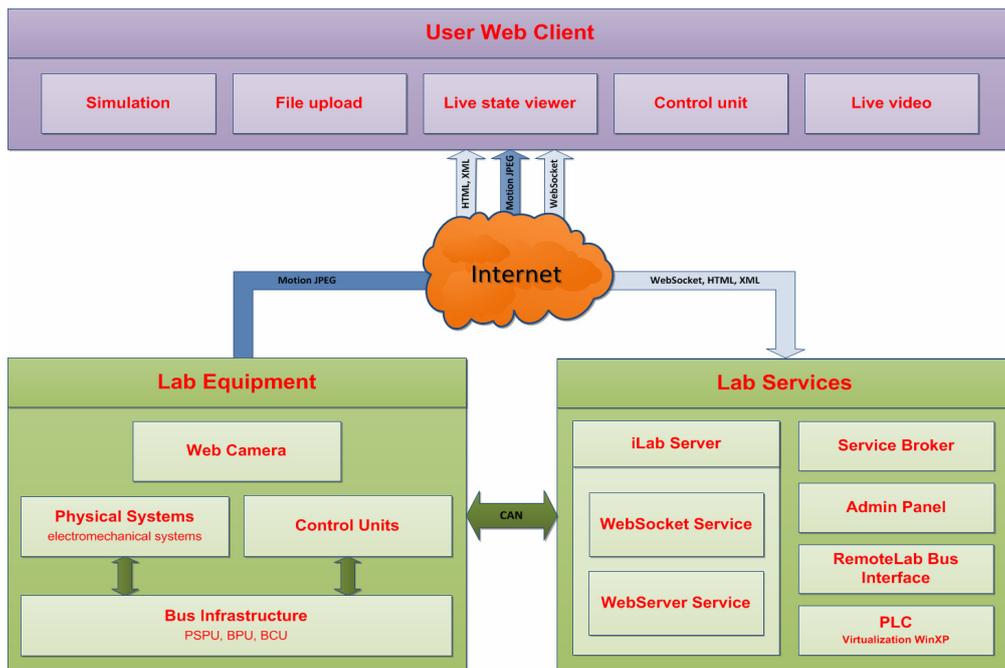


Figure 7. Schematic view of the remote lab components

Via the *Web-client* running on the student's PC at home the *physical system* (e.g., the electro-mechanical model of the water level control) can be controlled by using different *control units* (e.g., microcontroller, FPGA, PLC).

For the Web-clients modern Internet technologies like HTML5, jQuery and JavaScript can be used. These technologies make rich internet applications (RIAs) possible, which run in nearly any browser and on many terminal devices without needing any plugins.

These RIAs carry out many functions like animation, simulation, interpretation and control and therefore release the server from these tasks. The interaction with the user is speedup and can react faster and more direct.

By using this Web-interface, the student is able to:

- handle the experiment (e.g., start, stop, reset),
- change environmental variables if necessary and
- watch the experiment by manipulating environmental variables inside an I/O monitor or by observing the control of the physical system directly via a webcam.

Via an optional *local control panel* the student is able to manipulate the lab environment (e.g., the water level in the tank) or the actuators (e.g., the pump) when working on-site. Heart of this architecture is the *physical system protection unit*, which is described in detail in [15, 16].

A. Stand-alone Mode – Visual Prototyping

In case of using finite state machines (FSM) for specification, based upon an automaton graph, a student can use the JGIFT design environment [20] of the *REAL* system.

Figure 8 gives an impression of the verification and simulation features of the Web-based environment of the *REAL* system. The simulation model will be driven di-

rectly through the I/O signals by the control algorithm running on the embedded interpreter within the applet.

Assuming the student achieved a validated design, he gets the required next state and the output equations. By accessing the Web-browser interface of the *REAL* system, he is able to enter his algorithm (the received equations), handle the laboratory experiment (e.g., start, stop, reset) and change environmental variables if necessary. The control algorithm is executed by an interpreter running inside the student's client PC (e.g., implemented as a HTML 5 RIA). No Internet connectivity to the laboratory and the physical systems in lab is necessary for this mode (see Figure 9).

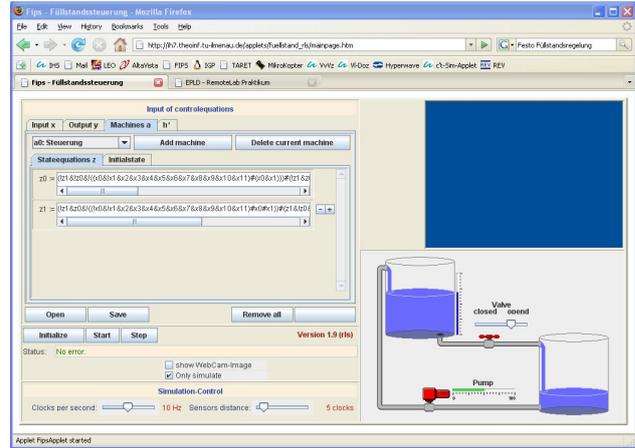


Figure 8. Offline regulation of the water level control (without Internet)

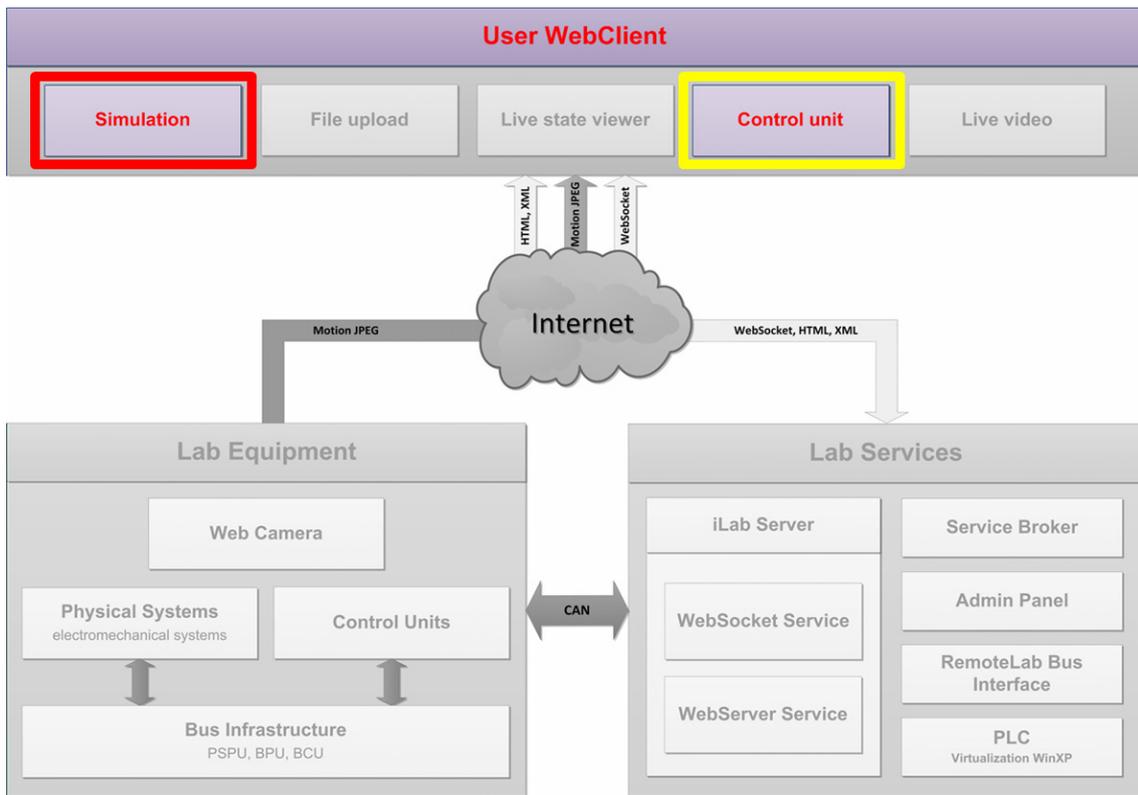


Figure 9. Stand-alone mode for visual prototyping

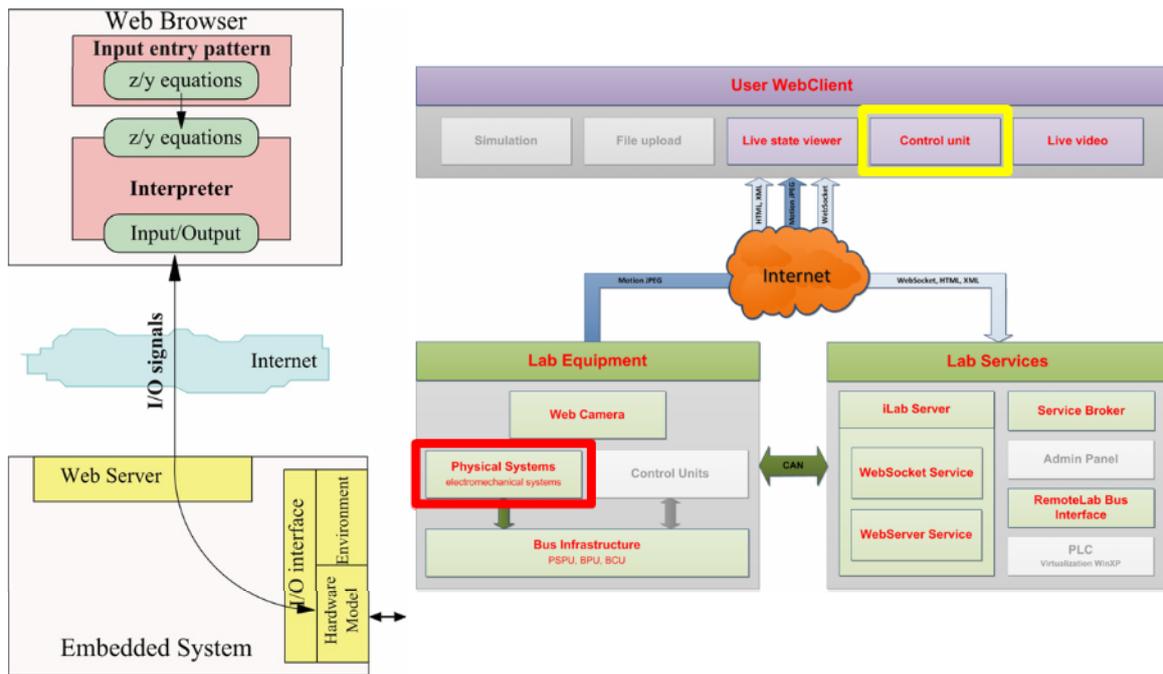


Figure 10. Remote control mode via a Web-client

B. Remote Control Mode – via Web-client

This operation mode is also for the FSM based specification based on equations. In this case, the physical system will be controlled via the Internet “from a distance” through the interpreter running inside the student’s client PC. No additional control units in the remote lab are necessary (see Figure 10). In this case, only the input and output signals of the physical system will be transferred via Internet.

An example of such a Web-client is shown in Figure 11.

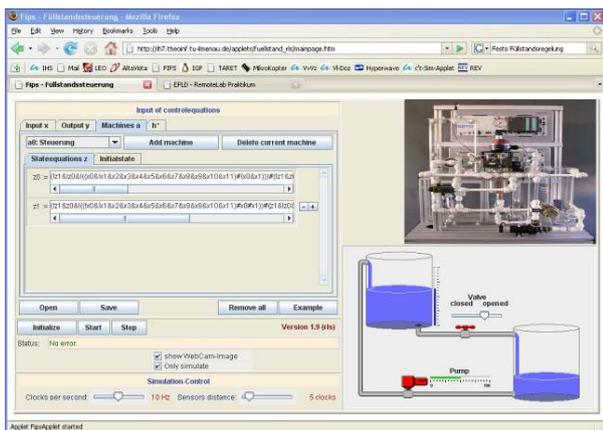


Figure 11. Online regulation of the water level control (with Internet)

C. Remote Control Mode – via Control Unit

This mode can be used to realize software or hardware oriented control tasks via connected microcontrollers or FPGAs as control units (see Figure 13).

Students can implement their control algorithm directly into a microcontroller for a software-oriented implem-

entation. Therefore, they use common (non-commercial) development tools, for example MPLAB IDE and/or C18 C-compiler by Microchip [21], or AVR Studio by Atmel [22], to develop Assembler- and/or C-coded software projects. After compilation, the generated software control algorithm is transferred via REAL Web-interface to the remote lab, where the hex code is programmed into the microcontroller (see Figure 12).

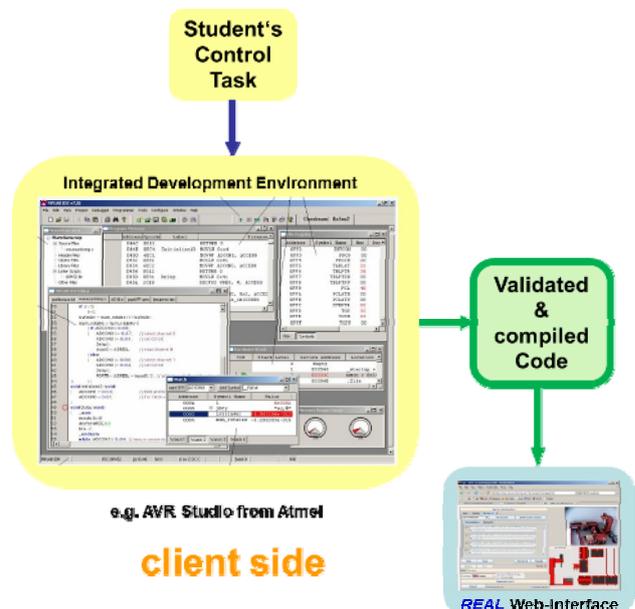


Figure 12. Software-oriented design of the control task

Now, the student can begin with his experiment, to check if his algorithm fulfills the requirements of the given control task.

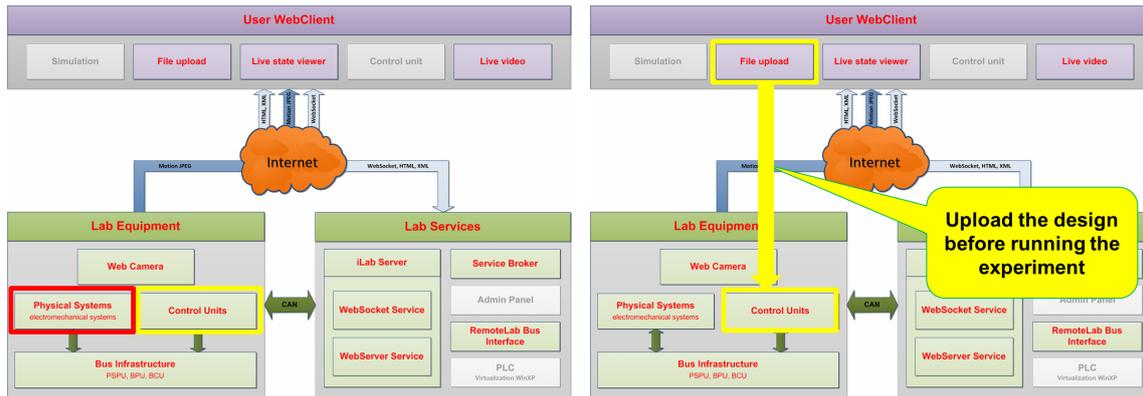


Figure 13. Remote control mode via a control unit

If a student prefers an exclusive hardware-oriented design using an FPGA and applying a hardware description language like VHDL as specification technique, he can prepare his design with common development tools, for example ISE, Quartus II, Diamond or others. The generated bit file is uploaded via the *REAL* Web-interface to the remote lab, where the FPGA will be programmed (see Figure 14). After programming the connected FPGA, the FPGA board operates as control unit for the designed control algorithm, and the student can start his experiment.

The student has the possibility to debug his design, by defining breakpoints depending on input/output signals from the physical system protection unit and stop the electro-mechanical model at certain sensor readings to validate the correct actor settings. It is also possible to do single step processing, by pausing the execution on every sensor/actor change.

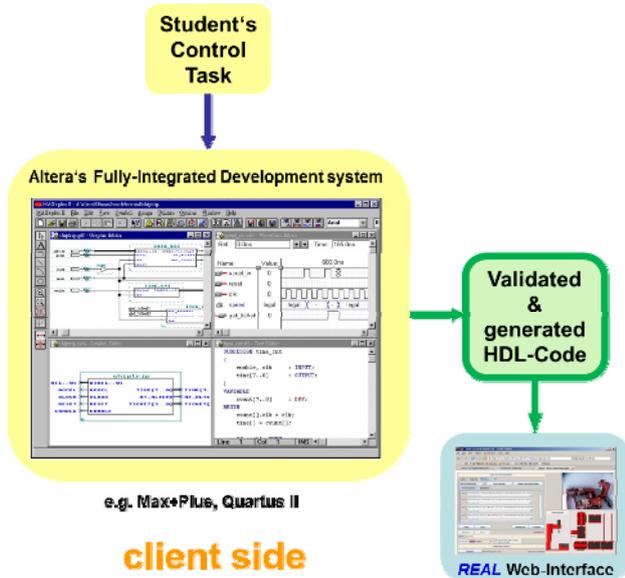


Figure 14. Hardware-oriented design of the control task

D. Virtual Control Mode – Visual Prototyping

This mode is comparable to the operation mode A (for visual prototyping). In this case, the simulation model is not connected to the interpreter, running inside the client,

but via the Internet to the real control units which are running inside the remote lab. In this case, the student can test his prepared software or hardware oriented design on the corresponding control unit (microcontroller or FPGA) without the need for a real physical system – before he will use operation mode C (remote control mode – via control unit) to test his design task on the physical system in the remote lab (see Figure 15).

Furthermore it is possible to have many students working on control algorithms for the same electro-mechanical model without disturbing each other. They can test and validate their design before connecting it to real hardware. This can eliminate the need to install multiple instances of the same electro-mechanical model in one lab and therefore reduce the costs of this lab.

Besides this, it is also possible to emulate electro-mechanical models that are not available in the lab.

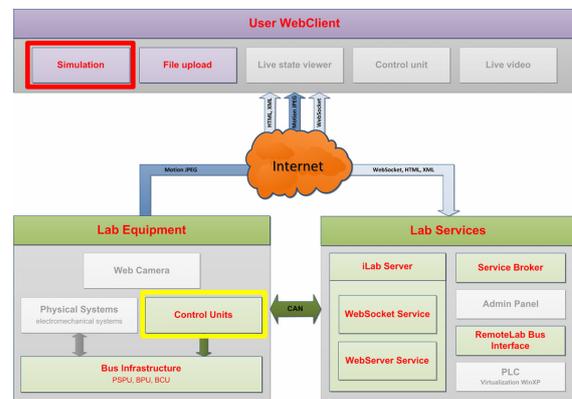


Figure 15. Virtual control mode for visual prototyping

E. Virtual Control Mode – Test Mode

This operation mode (see Figure 16) is mainly for debugging, testing and maintenance of the physical system protection unit.

By using this operation mode, it is possible to check the implemented reference design (as seen in Figure 4) in the physical system protection unit without the need to use a control unit or a physical system. This will be done by transmitting input and output pattern from the Web-client to the protection unit and by analyzing its response.

Because this mode is not preferential to execute lab experiments, it is not explained in detail.

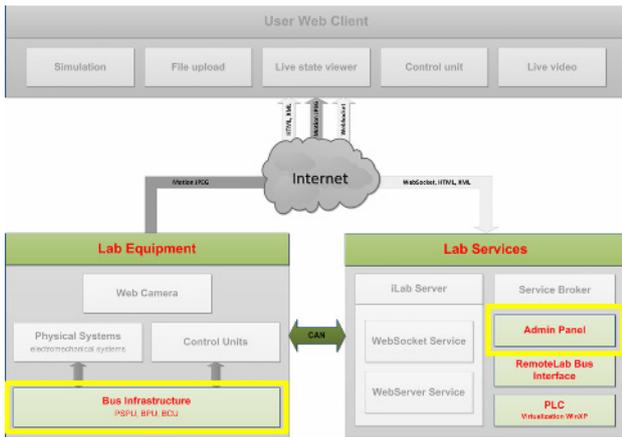


Figure 16. Virtual control mode to test the protection unit

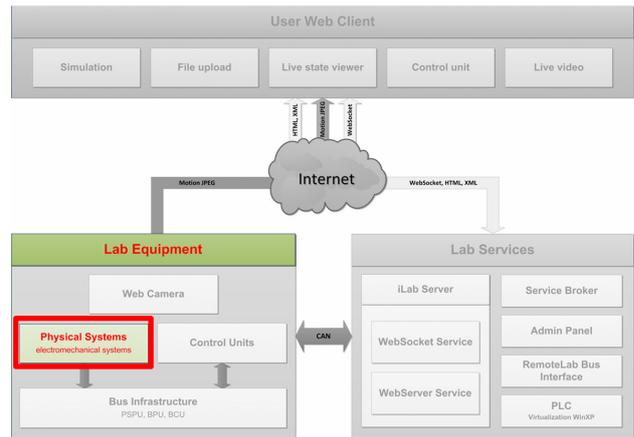


Figure 18. Manual local control

F. Local Control Mode – via a Control Unit

Besides the possibility to work in the lab remotely, the following two operation modes explain the usage of the *REAL* system for on-site experiments or demonstrations.

During an on-site lab experiment students can observe the whole hardware setup as well as all the physical system and environmental variables directly in the lab room. In this case they can check their control task (running on a connected control unit). They have access to the whole lab setup via a connected control panel. Figure 17 illustrates this operation mode.

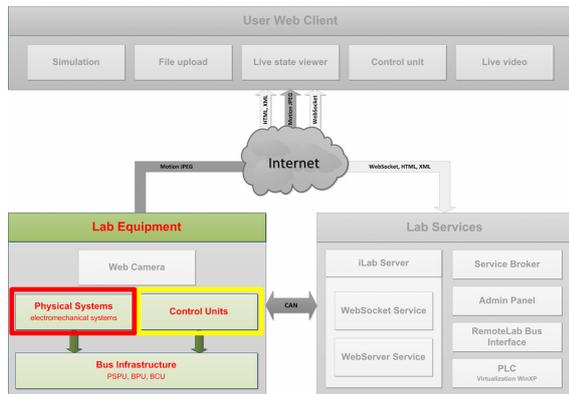


Figure 17. Local control mode via a control unit

In addition to its use for student experiments, this mode can also be used to demonstrate the remote lab on guided tours during open house presentations to inspire new students to study engineering courses.

G. Local Control Mode – manually

This operation mode (see Figure 18) can be used for demonstrations and maintenance of the connected electro-mechanical models.

By activating actuators (e.g., a water pump) manually without any control algorithm the sensor signals can be observed via the connected control panel.

This mode will also be used for hands-on usage of the demonstration system “labyrinth on ball balance plate” (see Figure 19) or can be used for testing and debugging the electro-mechanical parts of the physical systems for service and inspection.



Figure 19. Labyrinth model

We would like to provide this game-like experiment especially for promotional means for open house presentations at the university to attract new students in engineering disciplines.

H. Rapid Prototyping Mode

For this special operation mode a rapid-prototyping board for digital systems was developed, which is directly connected to the serial remote lab bus. In this case, no physical system protection unit is needed.

By using the *REAL* Web-interface, the student is able to

- upload his/her design,
- program the FPGA and
- handle the lab procedure.

The applet allows the student to manipulate all the inputs of the rapid prototyping board (slide switches, hex coding switches, and pushbuttons). He can observe the outputs of the board (7-segment displays, row of LEDs) virtually inside the Java applet. Figure 21 gives an impression of the applet’s Web-interface including a “photo” (image) of the rapid prototyping board.

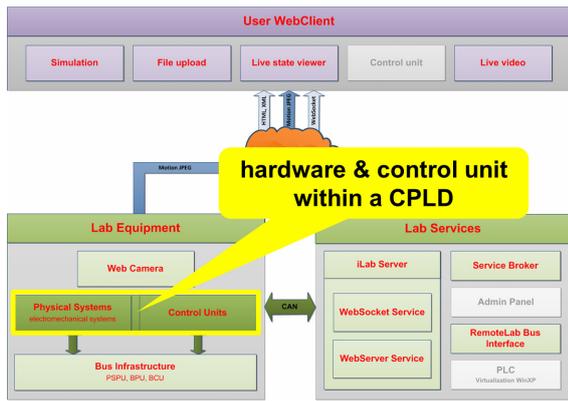


Figure 20: Rapid prototyping mode

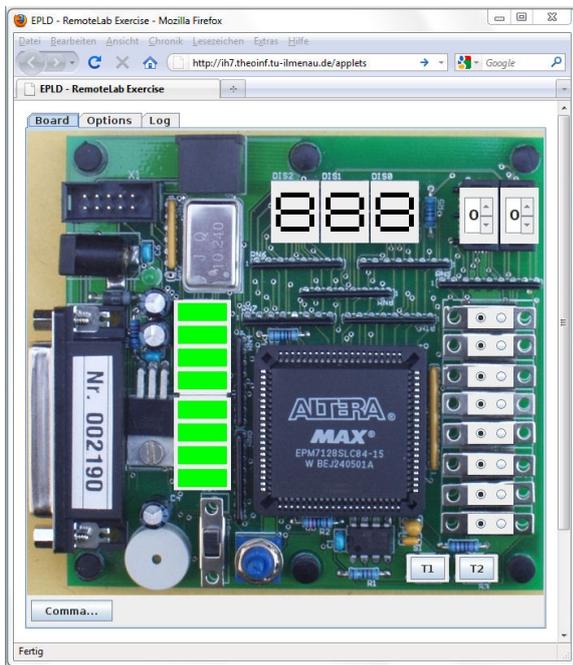


Figure 21. Web-interface of the Rapid Prototyping board

The manipulation of the input and output signals of the board are done virtually in the following way: For the look-and-feel of the applet, we use a “photo” of the board as background. All the inputs are realized as Java control elements and can be manipulated via the mouse interactively.

Changes are immediately sent to the rapid prototyping board and the corresponding results are displayed inside the applet. There are two general options to display the output results within the Web-interface:

- *Representation by Java components*
The graphical outputs are represented directly inside the “photo” using read-only Java components. Updates of the display only require information about the current state of the corresponding output signal. This enables students without fast Internet access to use the applet.
- *Webcam based feedback*
Another option is the use of a webcam to monitor the rapid prototyping board inside the lab room. The webcam image is replacing the background

“photo” of the applet. This allows the user to watch the results of his/her actions directly as if present in the lab. The overlaying Java display components are not used in this option and therefore invisible.

Both options can be configured using custom user settings in the corresponding tab of the applet. All actions are documented in a separate Log Tab.

By using this Web-based rapid prototyping board the following application fields are possible:

1) *Web-based Rapid Prototyping of Digital System*

By using common design tools (e.g. MaxPlus+, Quartus II from Altera [23]), the student is able to specify his design via

- *Text based design methods,*
Here the student can enter his design by means of logical equations, truth tables or hardware description languages (like VHDL or Verilog).
- *Graphically based design methods,*
The student can use block or schematic diagrams to input his design.
- *Integrated FSM editors.*
In case of sequential design tasks he can directly enter the derived automaton graphs with the built in FSM editor.

The editor itself generates VHDL code for the further design steps.

Finally, students will specify, describe, implement and verify different digital systems using “self-made” IP core libraries, e.g.

- Digital control systems
- Serial communication modules,
- Robot sensors,
- Model of a RISC processor.

Once the design is completed and error free, the student can test his solution using the described platform.

2) *Web-based Verification of Digital Systems*

Another use case of the rapid prototyping board is to identify the function of a given design (black box) or to find malfunctions of a well-known design. The CPLD will be programmed by the teacher, but the student has no ability to reprogram the device.

By manipulating the inputs of the unknown black box in order to find the malfunction (e.g. to enter all the input sets of a truth table or special input sequences), the student attempts to analyze the response (the real output signals of the board) to find out the function of the given design. The student has to enter his test vectors based on truth tables or input sequences one by one using the provided controls and observe the results.

It is planned to support the upload of a whole truth table or input sequence as a text file to the Web-interface of the prototyping board. In this case the student will be able to record the responses corresponding to the specified inputs creating a waveform file. This file can then be used for further investigation of the current design.

Visitor Mode

This mode of operation is intended for visiting the lab and passively taking part in experiments. It supplies a live video feed as well as a display of sensor and actor values from the electro-mechanical system of the selected experiment (see Figure 22).

Using the visitor mode does not require any login or booked time slot, but gives any interested person the possibility to view the lab and any running experiment. This mode is also useful for tele-teaching, where a teacher wants to present an experiment to a student via Internet.

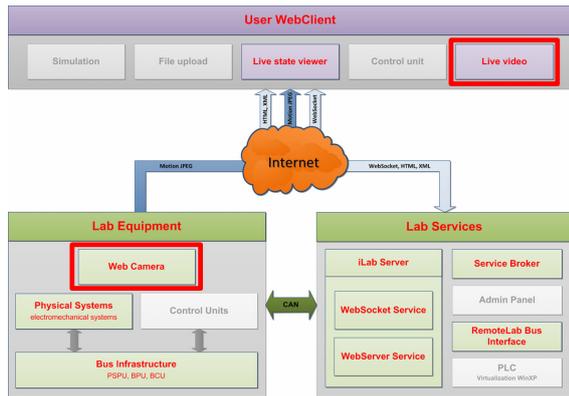


Figure 22. Visitor mode

IV. CONCLUSION AND FORESIGHT

We have discussed different operation modes of the REAL system for various fields of applications – based on a new flexible grid-based online lab structure. In addition to simulation-based experiments, on-side and off-side experiments with real physical systems can be offered for students as well.

Besides the features already mentioned in this article, even more functionality can be added using the new concept of having a Web-based protection unit that checks the user input against a reference model. This protection unit can be connected to a learning management system like “moodle” to forward any experimental results of the user. For an effective usage of the REAL system within learning management systems, the reference design and a method to check the student’s design against this reference design step by step will be traced by the LMS. Figure 23 shows this idea.

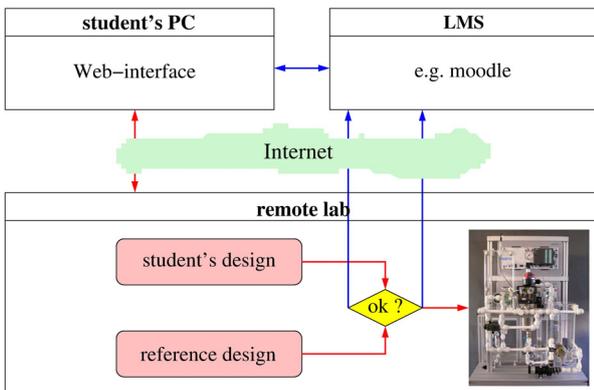
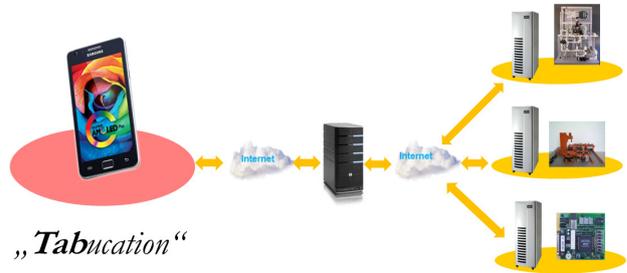


Figure 23. Observation of the student’s design under LMS control

The increasing capacity of wireless communication and the growing number of mobile devices (e.g. smartphones and tablets) on the one hand as well as modern Internet technologies like JavaScript, HTML5 and Web Sockets on the other hand provide new possibilities and challenges in the area of mobile learning (see Figure 24). In [24] a first Android client application for the iLab Shared Architecture is described.



„Tabucation“

Figure 24. Mobile learning by using modern wireless technologies

Our Integrated Communication Systems Group at the Ilmenau University of Technology is involved in different national and international e-Learning projects (e.g., [25]) in which it is increasingly necessary to allow and organize a shared use of equipment. That is why, the main focus of the REAL system is

- a Web-wide usage of different design tools and control units to control different physical systems in the lab room,
- a robust, fault-protected access to any connected physical system,
- an LMS-coupling for all control units and physical systems used in the remote lab as well as
- a worldwide interchange of online experiments with other universities by interconnecting the iLab Service Broker to an “iLab cloud” (see Figure 25), as proposed by the iLab Europe consortium, where we are involved, as well [26], [27].

All these requirements can be fulfilled using the concept and the infrastructure presented in this paper.

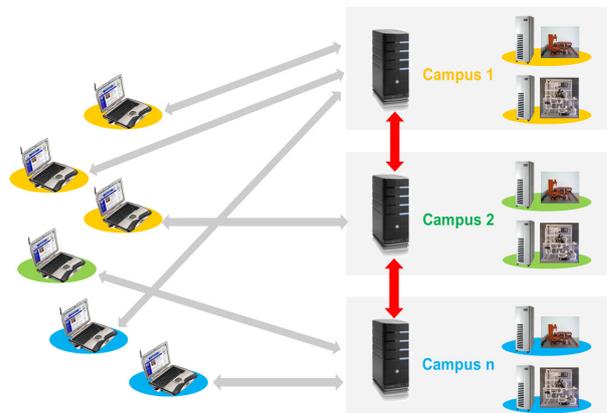


Figure 25. Interconnection of various iLab Broker to an “iLab cloud”

Future work will be concentrated on analyzing the student’s behavior during the experiments and developing an adaptive feedback and assessment system to support the students in a better way.

ACKNOWLEDGMENT

The authors would like to thank Silvia Krug, Stephan Simon and Stefan Vogel for their work within the *REAL* framework.

Parts of the *REAL* project are supported by the society "Friends of the Faculty of Computer Science and Automation". This work is also supported by the project "ICo-op – Industrial Cooperation and Creative Engineering Education based on Remote Engineering and Virtual Instrumentation" by the European Commission within the program "Tempus", Grant No 530278-TEMPUS-1-2012-1-DE-TEMPUS-JPHES [26].

REFERENCES

[1] S. Sire, F. Geoffroy and D. Gillet: "A Virtual Assistant for Sending Hints and Perturbations to Students based on an Electronic Laboratory Journal (eJournal)", Proceedings of the ITHET'03, Marrakech, Morocco, July 7-9, 2003.

[2] K. Henke and H.-D. Wuttke: "Web-based educational tool access", IASTED International Conference Computers and Advanced Technology in Education – CATE 2003, Rhodes, Greece, June 30 - July 2, 2003.

[3] H.-D. Wuttke and K. Henke: "Living Pictures – tool-oriented learning modules and laboratory via Internet", Proceedings of the ICEE-2002 International Conference on Engineering Education UMIST, Manchester, Great Britain, August 18-22, 2002.

[4] The iLab Project, from: <https://wikis.mit.edu/confluence/display/ILAB2/Home>

[5] WebLab Deusto, from: <https://www.weblab.deusto.es/web>.

[6] VISIR, from: <http://openlabs.bth.se/index.php>.

[7] LabShare Sahara, from: <http://www.labshare.edu.au/project/index.php>.

[8] REAL: "Remote Engineering and Applications Laboratory", <http://lanai.theoinf.tu-ilmenau.de/applets/index.htm>.

[9] Y. Torroja, et al.: "A Modular Environment for Learning Digital Control Applications", Microelectronics Education, Marcombo, S.A, 2002.

[10] T. A. Fjeldly, J. O. Strandman, R. Berntzen and M. S. Shur: "Advanced Solutions for Laboratory Experiments over the Internet", Engineering Education and Research-2001, Begell House Publishing.

[11] H.-D. Wuttke, K. Henke and N. Ludwig: "remote labs versus Virtual Labs for Teaching Digital System Design", Proceedings of the Int. Conf. On Computer Systems and Technologies CompSysTech'05, Varna, 2005.

[12] K. Henke, H.-D. Wuttke and T. Braune: "Virtual and remote labs in the Educational Process", International Conference on Remote Engineering and Virtual Instrumentation, REV2007, Porto, Portugal, June 25-27, 2007.

[13] K. Henke, H.-D. Wuttke and S. Hellbach: "Laboratory via Internet – new ways in education and research", Int. Journal of Computers and Applications, vol. 25, ACTA press, 2002.

[14] K. Henke, H.-D. Wuttke and T. Braune: "Rapid Prototyping Modules for Remote Engineering Applications", International Conference on Remote Engineering and Virtual Instrumentation, REV2008, Düsseldorf, Germany, June 23-25, 2008.

[15] K. Henke, St. Ostendorff and H.-D. Wuttke: "A Flexible and Scalable Infrastructure for Remote Laboratories - Robustness in Remote Engineering Laboratories", The Impact of Virtual, Remote and Real Logistics Labs - ImViReLL2012, Bremen, February 28 – March 02, 2012.

[16] K. Henke, St. Ostendorff, H.-D. Wuttke and St. Vogel: "A Grid Concept for Reliable, Flexible and Robust Remote Engineering Laboratories", International Conference on Remote Engineering

and Virtual Instrumentation, REV2012, Bilbao, Spain, July 4-6, 2012.

[17] MIT iLab Service Broker, Project homepage, from: <http://ilab.mit.edu/iLabServiceBroker>.

[18] H.-D. Wuttke, R. Ubar, K. Henke and A. Jutman: "Assessment of Student's Design Results in E-Learning-Scenarios", 8th Conference on Information Technology Based Higher Education and Training (ITHET2007), Kumamoto City, Japan, July 10-13, 2007.

[19] K. Henke: "Reusable Assessment Objects for Learning Management Systems", Computers and Advanced Technology in Education (CATE 2007), Beijing, China, October 8–10, 2007.

[20] JGIFT: "Java-based Graphical Interactive FSM Tools", <http://wcms1.rz.tu-ilmenau.de/fakia/index.php?id=780>, TU Ilmenau.

[21] Microchip Corporation, <http://www.microchip.com>.

[22] Atmel Corporation, <http://www.atmel.com>.

[23] Altera Corporation, <http://www.altera.com>.

[24] B. Deaky, D.G. Zutin and P.H. Bailey: "The First Android Client Application for the iLab Shared Architecture", International Journal of Online Engineering (iJOE), Vol 8, No 1 (2012).

[25] TRE – International Summer School in Technologies for Remote Engineering, www.ingenieria.deusto.es/summerschool2012.

[26] ICo-op project Website: <http://www.ICo-op.eu>.

[27] The iLab Europe Network Service Broker, from: <http://ilab-europe.net/iLabServiceBroker/>.

AUTHORS

Karsten Henke is with the Ilmenau University of Technology, Faculty of Computer Science and Automation, Integrated Communication Systems Group, 98684 Ilmenau, Germany, POB 10 05 65 (e-mail: karsten.henke@tu-ilmenau.de).

Steffen Ostendorff is with the Ilmenau University of Technology, Faculty of Computer Science and Automation, Integrated Communication Systems Group, 98684 Ilmenau, Germany, POB 10 05 65 (e-mail: steffen.ostendorff@tu-ilmenau.de).

Heinz-Dietrich Wuttke is with the Ilmenau University of Technology, Faculty of Computer Science and Automation, Integrated Communication Systems Group, 98684 Ilmenau, Germany, POB 10 05 65 (e-mail: dieter.wuttke@tu-ilmenau.de).

Tobias Vietzke is with the Ilmenau University of Technology, Master Student in Computer Engineering at the Faculty of Computer Science and Automation, 98684 Ilmenau, Germany, POB 10 05 65. He obtained his BSc. in Computer Engineering in 2012 (e-mail: tobias.vietzke@tu-ilmenau.de).

Christian Lutze is with the Ilmenau University of Technology, Master Student in Computer Engineering at the Faculty of Computer Science and Automation, 98684 Ilmenau, Germany, POB 10 05 65. He obtained his BSc. in Computer Engineering in 2012 (e-mail: christian.lutze@tu-ilmenau.de).

This work was supported in part by the European Commission within the program "Tempus", Grant No 530278-TEMPUS-1-2012-1-DE-TEMPUS-JPHES and by the "Friends of the Faculty of Computer Science and Automation" of the TU Ilmenau. It is an extended and modified version of a paper presented at the International Conference on Remote Engineering & Virtual Instrumentation (REV2012), held at University of Deusto, Bilbao, Spain, July 4-6, 2012. Received 01 March 2013. Published as resubmitted by the authors 20 March 2013.