A Low Cost Learning Environment for Collaborative Engineering

Dieter Müller[1], Frederico Menine Schaf[2]
[1] University of Bremen - artecLab, Bremen, Germany
[2]Universidade Federal do Rio Grande do Sul/Electrical Engineering Department, Porto Alegre, Brazil

***Abstract*—The widened access to collaborative learning environments and work assignments offered by remote labs and workshops is in the beginning stages of being established in the field of engineering education. However, remote lab equipment is often very cost intensive and not available for widespread use. In this paper, the application of inexpensive off-the shelf hardware and software is proposed, with the intent of setting up a collaborative learning environment for action-oriented PLC programming and configuration. PLC programming is regarded as a basic subject for control education. The objective is to provide engineering students ubiquitous access to real control hardware and devices in an uncomplicated and cost-oriented way.**

***Index Terms*—Computer Supported Collaborative Learning, Computer Supported Collaborative Work, Mixed Reality, Blended Learning.**

I. INTRODUCTION

The work presented in this paper is part of the research project CORELA (Collaborative Remote Laboratories). In CORELA we are developing concept and tools for learning communities and inter-site collaboration in which the stakeholders involved (e.g. vocational colleges, training enterprises, universities) can share and use distributed workshop and laboratory facilities via the Internet. This includes not only examples of fully remote experimentation and workshop settings in which all devices and equipment are real and available via the Internet, but also examples of mixed-reality environments in which real devices interact with simulation models and 3D Virtual Worlds [1]. The main application areas covered by CORELA are remote labs and workshops for mechatronics and distributed control systems (DCS).

One important objective of CORELA is to investigate how to make labs and workshops ubiquitous tools in engineering education. Remote resources are often not available for widespread use in educational environments (i.e. for technical and economic reasons). What are needed are solutions based on inexpensive hardware and software in place of high-cost commercial equipment.

In this paper, an example of an easy to use and cost oriented solution is presented, with which students can access a real programmable logic controller (PLC) in local settings or remotely over the Internet. The proposed solution illustrates the benefits and challenges of such an approach.

II. BACKGROUND

The presented concept idea in this paper is strongly inspired by two tools, called PROCESSING [2] and WIRING [3]. PROCESSING bases on an open project initiated by Ben Fry and Casey Reas [4] and evolved from ideas explored in the Aesthetics and Computation Group at the MIT Media Lab. PROCESSING comprises a programming language plus an easy to handle development environment. The programming language builds on the graphical capabilities of Java, but covers simplifying features and offers many new ones to develop interactive applications. WIRING itself directly builds on PROCESSING and is ideal for the prototyping of hard- and software components that augment physical objects with embedded processing and interaction capabilities. WIRING can also be used to develop stand-alone interactive objects or might be connected to software on another computer to build up client-server configurations. PROCESSING and WIRING are currently improved and maintained by a large user community. Both tools offer an open source programming environment (currently for Windows, Mac OS X, and Linux) and electronics prototyping platform based on flexible, easy-to-use hardware and software. There is a wide spectrum of hardware available, which is interoperable with PROCESSING and WIRING. Very popular is the Arduino [5] prototyping platform, which uses a small circuit board that includes a tiny microcontroller plus numerous I/O ports to connect with a wide range of sensors and actuators. It can be programmed very easily using the WIRING programming language.

WIRING and PROCESSING seem to fulfil the basic requirements for implementing a low cost learning tool and provide also hard- and software, which is necessary for a distributed virtual and real environment in the required application area.

Bencomo [9] lists about twenty requisites for successful implementation of remote experimentation environments. Among those, there are five basic requisites, which seem to be very essential, if the remote environment is to be used under everyday conditions on a university campus. These basic requirements are:

- Simple installation and use
- Access through the Internet
- No cost
- Interactivity and realism
- Total availability.

Last but not least, Buxtons' "less is more" design principle motivated us to reflect more about tiny and simple learning tools in relation to control-engineering education [10]. Keeping this principle and the above-mentioned requirements in mind, the idea of a low cost PLC lab kit is described in more detail in the following section.

III. System Design and Prototype Development

A. Overview

The general design idea is to build an experimentation lab kit for collaborative PLC training and control learning. The system is designed to integrate all necessary components either as a remote learning lab in faculty, or as an affordable standalone solution. Students should be able to take their lab home to learn independently, with minimal required equipment. From a didactical point of view, this is a very important aspect, because it motivates students to have all components available to build a complete system on the desk direct in front of them. Special care must be taken to achieve the necessary complexity while providing a single solution for lab and home use. Otherwise, solutions to tasks become trivial and students are not challenged.

The major design aspect of the proposed environment is to integrate it into existing virtual learning environments and into other experimental hardware and software. Lab components at work, school, and home should all be useable with a small amount of effort.

B. Basic structure

The proposed system can be divided into three major modules (Figure 1). These consist of the client interface, the communication and collaboration components, and the laboratory equipment. The system consists of a tiered architecture with three different layers, each layer providing its services to the next layer by using the services of the layer below.

A description of each component is given in the sections that follow. Figure 2 outlines the different layers with a brief description of their respective functions.

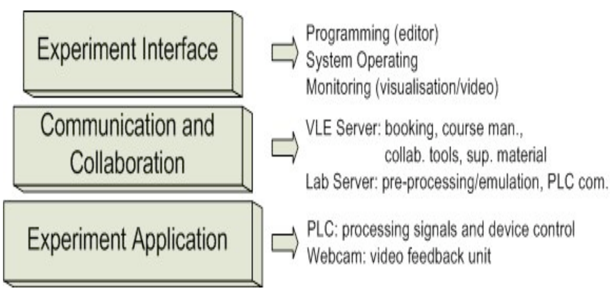

Figure 1. Architecture of the remote experimentation lab kit

1. Interface Layer The interface layer provides a graphical front end for the user. The system makes use of a standard Web browser. Users may access the lab at anytime from anywhere through a computer connected to the Internet (or Intranet). Through video streaming, the distant laboratory is made visible. The necessary client software plus the user interface will be implemented with PROCESSING.

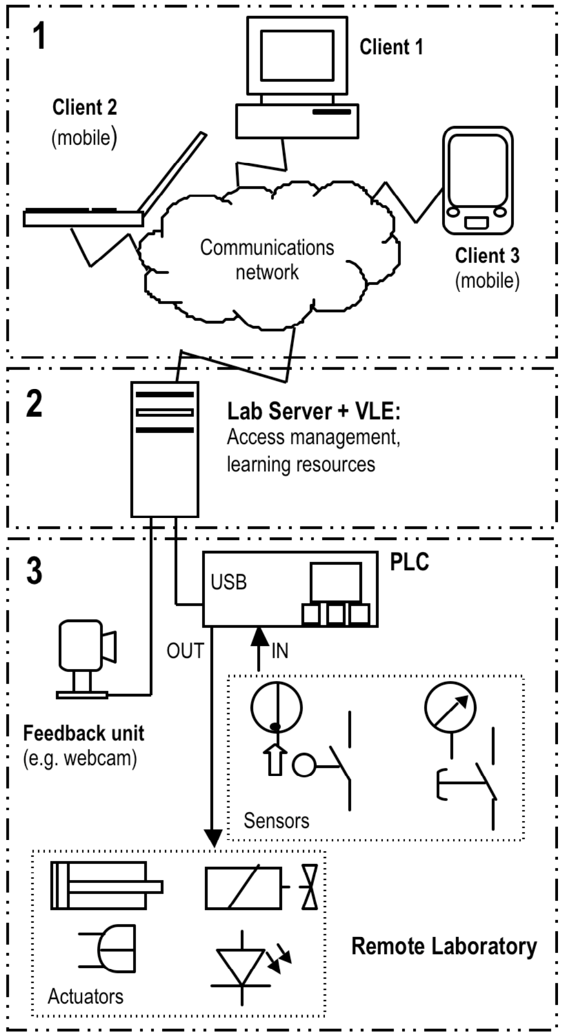

Figure 2. Schematic presentation of the remote PLC experimentation lab kit

2. Collaboration Layer The communication component builds a layer between the laboratory and the user. It consists of a VLE (Virtual Learning Environment) (e.g. Moodle [11]) and the lab server where the experiment is running. The lab server provides all modules for remote experimentation with a communication link (e.g. USB or Ethernet) to the PLC hardware. Placed in the actual learning environment of the faculty, these tools are supplemented by the virtual learning environment (VLE). The VLE plays a central role in the concept. It extends the PLC tools with learning material organized according to the faculty's need. Assessments and background material important for the experiment are also available. The VLE is also employed for synchronization with the available hardware (e.g. through a booking system). From a technical point of view, both services could be installed on one server.

3. Experiment Layer The actual laboratory consists of the automation control unit (PLC), real physical lab devices (can be also a complete functional plant) and a video feedback unit (e.g. webcam). The system is build on the WIRING compatible hardware Arduino [5] and will be implemented using the PROCESSING/WIRING programming tool kit.

C. *Implementation of a first prototype*

The current work is at early stages of development. Most of the results are in the research field. So that selected standards and technologies were studied. In a first step we have implemented a prototype, which bases on the PROCESSING programming tools and a WIRING compatible hardware controller. For the hardware we choose the Arduino board, which is widely available and very cheap.

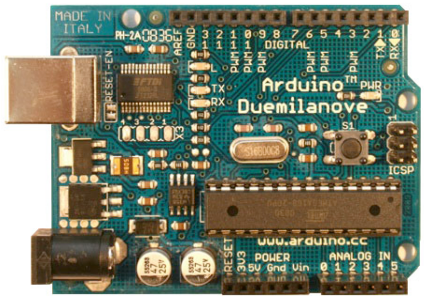

Figure 3. Arduino board (Duemilanove model).

The Arduino board consists of an 8-bit Atmel AVR microcontroller with complementary components (e.g. I/O ports, USB connection) to facilitate programming and incorporation into other hardware infrastructure (Fig. 3). The Arduino's microcontroller is also pre-programmed with a bootloader that simplifies uploading of programs to the on-chip flash memory, compared with other boards that typically need an external chip programmer.

Step7 Instruction List (AWL)
ORGANIZATION_BLOCK OB1
U M0.0
= A0.6
UN A0.7
UN E0.7
SPB PB1
ErrChk:
U A0.0
U A0.1
R A0.0
R A0.3
S A0.7
U E0.7
R A0.7
BE

Table 1. Programming example.

The developed prototype includes a basic Web service providing a compiler for PLC languages and an interface driver to connect the PLC controller board with the lab server. The PLC compiler itself is written in PROCESSING and is capable to translate a PLC instruction lists in an intermediate runtime code, which could be executed on the PLC controller board hardware. Currently a subset of STEP 7 AWL [28] is implemented (see Table 1). The concept is open for further enhancements.

Clients' access is realized through TCP/IP network connections. We used the PROCESSING network library to implement the appropriate services. Figure 4 shows the flow of information between client, lab server and PLC. After connecting to the server, compiled PLC programs could be sent to the server. The server forwards the code to the connected PLC controller hardware and provides feedback on errors or success. PLC programs could then be started on the real PLC via the Web service (Fig. 5).

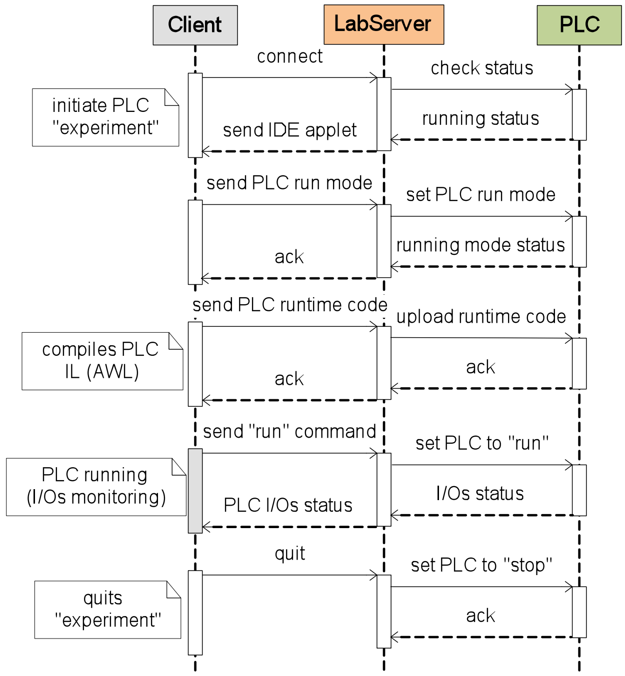

Figure 4. Sequential diagram of the client to PLC information flow

A challenge is updating the PLC while a program is running. Normally, the starting and stopping of a PLC program is done through a switch on the hardware controller. As this is not feasible remotely, our solution is to provide an additional framework that listens to the lab server - PLC connection. When an event on the connection line is received, the running PLC program is interrupted and the internal memory and/or run mode are updated. Afterwards, the PLC is soft restarted by calling its setup procedure.

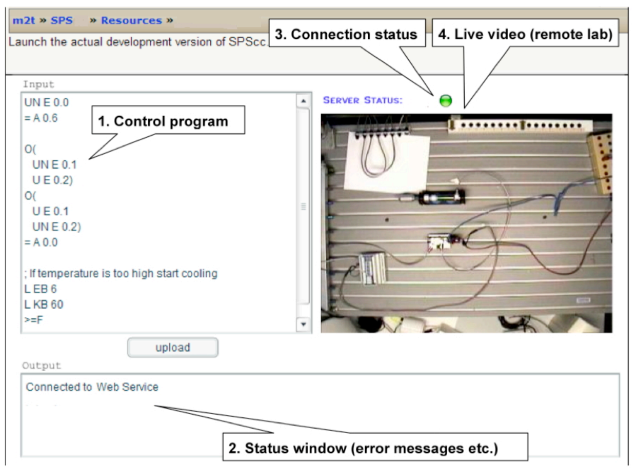

Figure 5. User Interface

Following the idea of PLC easy access and virtual learning environment integration, a simple user interface in PROCCESSING was developed (Fig. 5). FESTO [22] components for setting up an e-pneumatic circuit, which is remotely controlled by the PLC were employed. There is no need to stick to this certain interface, own clients using the comfortable PROCESSING framework and libraries could be implemented. Currently, a solution to link the PLC to a virtual factory (3D client), which is modeled in OpenSim [12] (Fig. 6), is under development.

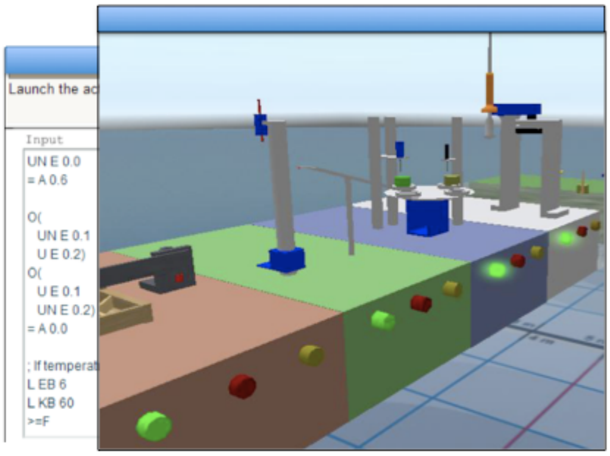

Figure 6. OpenSim 3D client.

IV. Related Research

The presented solution in this paper is influenced by experiences with a PLC controlled solar plant, evaluated for vocational training in the MARVEL project [13]. The system consists of several Siemens PLCs in combination with Siemens WinCC on a complex industrial bases [14]. Other examples relevant within the context of this paper come from the Control-Net project, which developed remote lab examples for plant and process control with STEP7 [15]. Langmann introduced a remote PLC lab, which provides a programmable authentic assembly station [16]. This lab is designed for control engineering courses based on IEC 61131-3. Hsieh et al. [17] introduced an online programmable logic controller learning system based on a Soft PLC. Sousa and Carvalho [18] have implemented an IEC 61131-3 IL (Instruction List) and ST (Structured Text) compiler that is part of an open source PLC project called MatPLC [19]. Shen et al. developed low-cost modules for remote engineering to perform lab experiments over the Internet [20].

In general, there is an interesting discussion in control engineering research about cost oriented solutions, not only for educational, but also for industrial applications [7][8]. This motivates our work as well. While research in the control-engineering field is essential, we also got impulses from software engineering, especially approaches to interconnecting middleware and control devices by Web services, like SOAP [27].

V. Broader Impact of work

The goal and subject matter of our work is a contribution to a construction kit for automation and control education. The history of physical and computationally enhanced construction kits suggests that these tools have a very motivating role in engineering and science education [21]. Existing construction kits like *LEGO Mindstorms* [24] or *Fischertechnik* [23] are typical and prominent examples of such type of toys, used not only in schools, but also in universities and industry for supporting creativity and encouraging design skills. While these kits can be used to create dynamic machines, affordable soft- and hardware for embedded computation in combination with both powerful software applications, like PROCESSING and WIRING, open further perspectives to build up distributed control systems (DCS). As PROCESSING and WIRING offer interoperability with external tools, like LabView [26] or MatLab [25] in combination with *LEGO*, *Fischertechnik* or other available physical construction tools, this in turn offers the ability to develop next generation learning environments, suitable to use in control engineering education.

Distributed control systems differ from traditional control systems: there is not only the sensing and processing of information, but also a focus on cybernetics, as a closed loop technology for action and reaction. This requires new engineering thinking, skills and perspectives. An every day world, full of sensors and actors, controlled not only by distributed local devices, but also by central computers, requires the engineers to think in new categories of time, location, and space. Although in automatic control we are used to apply distributed algorithms via fast network connections and protocols like Profibus, the open ended situation of every day surroundings poses a new challenge for adequate and adapted reaction times, suitable not only for deterministic automata and machines, but also for human beings and natural objects. Distributed automation devices will be carried around in the future. Physical and social contexts are changing in an arbitrary way and have to be foreseen in some engineering sense. Awareness of privacy, safety, and security has to be considered in the design and operation of these systems. All these topics are of course subjects for good engineering education, but in collaborative engineering they get a new relevance.

To support an easy integration of these topics into everyday teaching of mechatronics and automation technology, we need more low cost devices for students to use from home and at their own places, from lecture rooms connected to laboratories, or vocational schools connected to students industrial workplaces. Our PLC solution is intended to contribute to this aim.

VI. Conclusion

The proposed prototype enables ubiquitous access to real control hardware and devices in a very uncomplicated and cost oriented way. Our aim is to demonstrate that such a simple system can support active learning within distributed and collaborative learning settings. The prototype is a very first step and additional features (e.g. collaboration facilities) are possible through external applications (e.g. Skype, Moodle). The interoperability with other engineering tools, like LabView or Matlab/Simulink is abound.

The tool challenges students by providing a highly dynamic and reconfigurable setting. Learning tasks can be developed for different levels of complexity without changing the general software structure. It also integrates into existing hard- and software as well as learning environments. The system is completely based on open source software. The off-the-shelf PLC hardware used costs less than 30 Euros.

REFERENCES

[1] Schaf ,F. M., Müller, D., Bruns, F.W., Pereira C. E., Erbe, H.-H. (2009). Collaborative Learning and Engineering Workspaces. *To be appear in Annual Reviews in Control.*

[2] Processing: http://www.processing.org

[3] Wiring: http://wiring.org.co

[4] Reas, C. and Fry, B.: Processing (2007): A Programming Handbook for Visual Designers and Artists. 2007, MIT Press.

[5] Arduino: http://www.arduino.cc/

[6] Bernhardt, R. and H.-H. Erbe (ed.) (2001). *Cost Oriented Automation*. Proceedings of 6th IFAC Symposium, Berlin, Germany, October 8-9, Elsevier Science Ltd., Oxford.

[7] Wollherr, W. and M. Buss (2003). *Cost-oriented virtual reality and real-time control system architecture*. In: Robotica (2003), Cambridge University Press, Cambridge, UK, Vol. 21, pp. 289-294.

[8] Rasche, A., B. Rabe, P. Tröger and A. Polze (2004). *Distributed Control Lab*. Proceedings of The 1st International Workshop on e-learning and Virtual and Remote Laboratories (VIRTUAL-LAB′2004), Portugal, pp. 150-160.

[9] Bencomo, S.D. (2004). Control Learning: Present and Future. *Annual Reviews in Control 28 (2004),* pp. 115-136.

[10] Buxton, W. (2001). Less is More (More or Less), in P. Denning (Ed.), The Invisible Future: The seamless integration of technology in everyday life. New York: McGraw Hill, pp. 145-179.

[11] Moodle: http://moodle.org/

[12] OpenSim: http://opensimulator.org

[13] Müller, D. and J. M. Ferreira (2004). MARVEL: A mixed-reality learning environment for vocational training in mechatronics. In *Proceedings of the Technology Enhanced Learning International Conference (TEL'03)*, Milan, Italy, pp. 65-72.

[14] Hoell, W. and H. Martens (2005). Remote Process Control of a full-scale Solar Heating System. Learning Environment Design, Course Notes, and Evaluation. *MARVEL final report*. Bremen.

[15] Meyer, J.-T., A. Georgiadis and S. Schwarz (2004). *Online Hands-On Trainings – Real Worlds in Virtual Environments*. In Proceedings of the EDEN Annual Conference, Budapest.

[16] Langmann, R. (2004). E-Learning & Doing - A Remote Lab Network for the Education on the Engineering. In Proceedings of the TELEC, Santiago de Cuba.

[17] Hsieh, S.-J., P. Y. Hsieh and D. Zhang (2003). Web-Based Simulations and Intelligent Tutoring System for Programmable Logic Controller. ASSE/IEEE Frontiers in Education Conference Proceedings, Boulder, CO, November 5-8

[18] de Sousa, M. and A. Carvalho (2005). Programming with the IEC 61131-3 Languages and the MatPLC. The Industrial Information Technology Handbook, CRC Press, 2005, pp 1-21

[19] MatPLC: http://mat.sourceforge.net

[20] Shen, H., Shur, M.S., Fjeldly, T. A. and K. Smith (2000). Low-Cost Modules for Remote Engineering Education: Performing Laboratory Experiments over the Internet. In Proceedings of 29th ASEE/IEEE Frontiers in Education Conference (FIE'00), Kansas City, Missouri.

[21] Schweikardt, E. and M. D. Gross (2007). A Brief Survey of Distributed Computational Toys". DIGITEL 2007: The First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning, Jhongli, Taiwan.

[22] FESTO. http://www.festo.com

[23] Fischertechnik. http://www.fischertechnik.com

[24] LEGO: Mindstorms. http://mindstorms.lego.com

[25] MatLab: http://www.mathworks.com

[26] LabVIEW: http://www.ni.com/labview

[27] SOAP: http://www.w3.org/TR/soap/

[28] Step 7: Siemens SIMATIC. Programming with STEP 7 V5.4. Manual. Nurnberg 2009.

AUTHORS

Dieter Müller is with the Art-Work-Technology Lab (artecLab), University of Bremen, Enrique-Schmidt-Strasse 7 – D-28359 Bremen - GERMANY (e-mail: mueller@artec.uni-bremen.de).

F. M. Schaf is with the Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, RS, BRAZIL (e-mail: fredms@ece.ufrgs.br).

Manuscript received 27 May 2009. Published as submitted by the authors.