

ENHANCING THE INTERNET-BASED CONTROL IN TEACHING LABORATORIES USING SOFTWARE AGENTS

Mihaela Albu¹, Florin Mihai¹, Cătălin Buiu², Denis Gillet³

1-Department of Electrical Measurements, "Politehnica" University of Bucharest, Romania, albu@electro.masuri.pub.ro

2- Department of Automatic Control and Systems Engineering, "Politehnica" University of Bucharest, Romania, cbuiu@yahoo.com

3- Institut d'automatique (IA), Swiss Federal Institute of Technology, Lausanne, Switzerland, denis.gillet@epfl.ch

Abstract: This paper presents a laboratory for remote automatic control via Internet. This distributed laboratory (DLAB) is located at POLITEHNICA University of Bucharest (PUB). The control architectures are based on advanced communication technologies, distributed measurement systems, and virtual instrumentation. Software agents are assisting the users in performing their activity. The results obtained so far strongly suggest that remote collaborative experimentation mediated by software agents is an efficient and necessary tool of a new era in control engineering. *Copyright © 2001 IFAC*

Keywords: remote control; autonomous mobile robots, agents.

1. INTRODUCTION

New teaching developments (WWW, applications of virtual reality) are having dramatic effects on learning environments at all levels. This leads to the development of new teaching and learning styles. Among these, learning based on electronic materials, on-going interaction between partners in the learning process and collaborative learning exist already in virtual classrooms, as Virtual Learning Environments (VLEs) but are still very far from being effective in electrical engineering instruction laboratories.

We should firstly discriminate between *generic VLEs* and the *specific VLEs* - which are those VLEs dedicated to electrical engineering studies which encompass *remote experimentation* (Gillet, 2001) and a virtual laboratory. In this case, we consider *simulation* (an important component of both types of VLEs) as a key issue as long as we interpret the

scientific meaning of simulation as "goal-directed experimentation with dynamic models" (Oeren, 2001). On the other hand, it is important that the special feeling of engineering experimentation is preserved in the lab.

We will discuss the application of artificial intelligence techniques in the monitoring and control of experimental systems in electrical engineering. Data acquisition, monitoring and control systems are developed in order to perform these tasks. Artificial intelligence techniques will play an important role in this context.

Among these techniques, that of software agents (SA) are promising to become an ideal tool for higher education, providing it with discretion, efficiency, inflexibility, reliability and a major quality: they can be continuously tested, improved and, when not corresponding, dismissed.

In our opinion, the future belongs to group learning environments where students learn from each other and teachers (in real- or virtual form), and by interacting with computer-supported simulations. The project in course at PUB (DLAB) aims to explore a genuine zone in the cyberspace, preserving the scope of engineering education against the temptation of losing the Self into the Cyberculture. Both students and educators have to be helped, as humans are no more the optimum guides in such a hyperspace.

2. REMOTE EXPERIMENTATION OVER INTERNET - A LEARNING CORNER INTO THE CYBERSPACE

Learning is not simply a matter of individual development, it is also by definition a social activity. It takes place in and through situated social activities and practices. In this paper, we will consider learning as it emerges from interactions within the virtual environment, as seen by the software agents and from interactions between the virtual environment and the external environment(s) in which learners and instructors participate (i.e. laboratory).

The dominance of the Internet in the development of information and communication technology has made *Web-based distributed solutions* increasingly attractive. Apart from providing other services, the World Wide Web is being looked upon as an environment for hosting modeling and simulation applications (Guru *et al.*, 2000). One of the major advantages provided by simulation is its ability to help trainees develop technical skills. The student uses virtual hardware (a simulator) to develop skills for manipulating the corresponding real world system.

Online simulation is simulation running concurrently with the real world system. There are three major application areas for online simulation: to support the operation of a real world system; to support on-line diagnostics, and to augment reality.

In the case of *augmented-reality simulation*, real and virtual entities (humans or equipment) as well as the environment can coexist at the same time and, therefore, operations can take place in a richer (augmented) environment.

Cyberspace, an expression first coined by novelist William Gibson, and now used when referring to the Internet, is a *world* in which computers and people continuously interact. If we accept a broader concept of interactivity on the Internet, we can offer an extremely expanded image for a future that includes telecommunications, and many other media. We refer to this future image of telecommunications as *hypercommunications* (Nakatsu, 1998).

3. MULTI-USER APPLICATIONS FOR TEACHING CONTROL ISSUES

At this time, simple control experiments are available worldwide for remote experimentation. Our research objective is to implement remote control architectures for more complex real world processes, for example highly non-linear processes. That opens the way for intelligent control algorithms in remote experimentation. Diagnosis, alarm handling and user assistance tools extracting information from a common knowledge base are examples of the facilities that can help to solve problems when the human experts are not available (Talukdar, 1999).

The real-world *Distributed LABORatory* (DLAB) for *remote experimentation and control* hosts a total of eight PC-stations type AMD K7-600 DURON, and a Network-server type AMD K7-800, interconnected via a fiber-optic link with the main node of the Romanian Academic *RoEduNet* WAN.

One set of client applications encompasses remote-access virtual instruments, mainly digital multimeters and oscilloscopes for measuring and monitoring purposes, and function generators for test-signal generation. Any user can use any virtual instrument. A chat window allows fast communication between all users (one of them being the instructor).

Two different types of DAQ cards are presently being used, namely the high-performance National Instruments AT-MIO-16, and the Keithley PCI-MIO-16E-4. *LabVIEW™* is used for programming both DAQ card types. *LabVIEW™* features interactive graphics, a state-of-the-art user interface, a powerful graphical programming language, and is supported by extensive libraries (Salzmann, 2000).

Students can practically experiment with all typical I/O functions like analog input (A/D conversion), buffered data acquisition (high-speed A/D conversion), analog output (D/A conversion), signal generator, digital I/O, counter/timer operations, etc.

They can also practically experiment with a fuzzy-logic controller. For this purpose, an industry-standard, stand-alone FLC, type Kloeckner-Moeller *fuzzyPLC*, Model PS 4-401-MM2, is being used. Standard PLC control software together with a SUCOsoft *fuzzyTECH-40* software package allows students to create real-world fuzzy-logic applications. The local application server or any of the remote client PCs can serve as an *engineering station* for designing and testing control algorithms.

The local application server's *visual interface* (VI) enables communication between the DAQ card, *fuzzyPLC* and remote client applications via both TCP/IP and multi-port-access protocols. It is always hosted by the local application server while the remote client's VI can be hosted by his PC or

downloaded from a Web page. The application server VI also registers the number and main properties of all *remote-experimentation* participating (logged-in) clients.

Working on the power plant's application server (local client) or in the DLAB (remote client) they first design control algorithm modules and then download them into the *local* control hardware (DAQ/FLC) for *in-process* testing and optimization. This kind of approach allows students to work in a real-time, real-world environment, i.e. in an active power plant (Buiu *et al.*, 2001). They have been given a means to work (and in a positive sense, to "play") interactively in a *remote experimentation environment*.

The physical layout of DLAB-control also has two Khepera robots and several devices controlled via parallel (Centronics) interface. One of them is a plug-in device, with a simple schematics using relays, which enable the remote students to put in a shut-down mode all devices (such as measurement instruments, fuzzy-PLC systems etc.) connected to the computers network via a digital interface.

3.1 Signal-processing application

Fig. 1.a depicts an example for multiple use of a *National Instruments* type DAQ card and a *Hewlett-Packard* Function generator controlled by a remote client PC.

The control result can be monitored in the bottom-right window. The video images from real instruments are taken by a video camera mounted on the server-side.

One can see that are 2 users for each client application (controlling the function generator and the DAQ-card). The server applications may be shared by multiple client-applications. The image taken with a video camera enables remote-user to adjust the application settings (as the scan rate and the number of acquired points in figure 1c) as well as the camera position (through the computer-driven step-by-step motor).

All instruments are connected to a PC (where a corresponding server application is running) via a serial or GPIB interface allowing an user identifier (logg-on name, a logo etc.) to be locally displayed (as we can see in figure 1b). This is one of the "signs" that the experiment is not another level of simulation (possibly to be done up to re-build the simple reality of a control laboratory), yet a real-world experiment shared by learning participants into a larger, simulated virtual environment.

The chat window is designed for three- different user modes: an "all-to-all mode", a "student-teacher" mode, as a private communication channel and a "spoken" mode, when the in chat window written strings are delivered to *Ash*, the spoken (local) agent.



(a)



(b)



(c)

Fig. 1: Panel-windows of a multi-user application on Internet: a, b, c - multiple use of DAQ-card and function generator; d-example of an appropriate selection for the DAQ-card settings.

The camera used for video broadcasting is a simple (on USB), fixed PhilipsTU. In order to be able to see the on the experiment side lab-images, the camera was fixed on a step-by-step TEAC 6-coils motor, controlled via the parallel (Centronics) interface through a 2803 Darlington array was used. The motor is easily controlled under LabVIEW (a dedicated client-server application).

3.2 Mobile robotics application

One of the experiments is using a Khepera robot (connected to the PC via a serial port), a PC (parallel port) driven step-by-step motor, and a video camera (on USB). The control algorithm is written in Matlab and it's meant to drive the video camera (attach to the step-by-step motor) in order to best follow the moving robot on its trajectory. The control has also to take into account two user-defined parameters, i.e. the distance between camera and robot and camera angular position, which are defined as relative parameters.

The second mobile robotics experiment allows the users to test different Braitenberg vehicles. In his book "Vehicles: Experiments in Synthetic Psychology", Valentino Braitenberg describes a series of thought experiments in which "vehicles" with simple internal structure behave in unexpectedly complex ways (Braitenberg, 1984). This experiment addresses the problem of collective robotics from a tele-presence point of view.

First of all, a user has the option to remotely test simple robot commands (turn left, back, forward etc.) (fig. 2).



Fig. 2. Simple robot commands

Secondly, the remote user has the option to select from four possible behaviours:

1. aggressive robot;
2. loving robot;
3. coward robot;
4. explorer robot.

and to remotely test each behaviour (fig. 3).

A video window is always displayed on the screen, giving the user an idea of how looks that particular behaviour. The Braitenberg-like control algorithms, which simply relate the motors of the robot to the sensors, are written in Java. This experiment is used by students in automatic control and by students in cognitive science, as well.

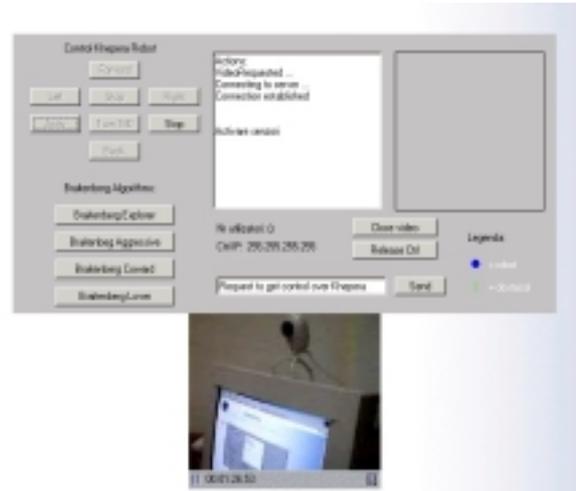


Fig. 3. Collective tele-robotics experiment (main window)

Two experimental set-ups are possible. In the first one, there's only one remote student testing a particular behavior, while in the second one a group of maximum three students (there are 3 physical robots) may test a particular behavior for each robot. So, we have a team of 3 robots with different individual behaviors and a global, complex behavior. Each use is able to perceive the global behavior in its browser's video window. There's also a supervisor of the whole experiment, that is the teacher that has a global video image of the experiment also and the opportunity to interact with each student by using a chat window.

4. THE AI SILENT VOICES IN DLAB

The traditional classroom is designed as a kind of broadcast medium, a stage for the instructor's presentation of knowledge ("sage-on-the-stage"). In the networked classroom, however, emphasis falls on the processes by which students engage with one another, with the instructor, with the course material, and with the virtual environment itself, in an on-going process of creating new knowledge ("guide-on-the-side").

In the same time, the virtual environment actually permits to the instructor more substantial engagement with more students than is generally possible in the traditional classroom. We choose to use software agents in order to assist students in learning how to use the equipments in the laboratory and to have an on-line help for the software we designed.

An agent is the fundamental actor in a domain. It combines one or more service capabilities into a unified and integrated execution model that can include access to external software, human users and communication facilities.

The client application has equally a chat window (seen only by the students that are running the experiments) and a video window for following the robot path. Two agents will split the tasks in assisting the student when running the application: one of them, *Aker* (Agent for Khepera Robot), will give "advices" at the robot control level and the other one, *Ash* (Agent for Spoken Help), will assist the student in running the client application, providing on-line help (even spoken) which is structured on different levels of explanations

Firstly the Microsoft Agent (*MSAgent*) technology was used, when considering all the applications running exclusively under Windows OS. Along with the "translation" facility offered by LabVIEW for the Mac-users, we are designing the agents for independent platforms.

5. CONCLUSIONS AND FURTHER WORK

A real-world *Distributed Laboratory* (DLAB) for *remote experimentation* has been set up at the Electrical Measurements and Automatic Control Departments of the Polytechnic University of Bucharest and practical *remote experimentation* tests have become a firm part of latest curricula.

The idea behind the laboratory is the versatile use of the equipment together with on-line testing of control algorithms written by the students. This led to two different approaches, depending on the effective users: the students that actually inhabit the laboratory during classes or the remote (in sense of other campuses) users that access the equipment via Internet. Both solutions pretend multi-user, client-server architecture of the applications and allow complete data management. However, major differences appear when considering the speed, the limitation in bandwidth and, more critical, the multiple computing platforms. The aim of DLAB is to be part of a ring of remote-experimentation and control teaching laboratories, spread in Romania and Europe also.

LabVIEW™ was used for developing existing server and client applications. Java was used to implement the remote Braintenberg vehicles experiment. We plan to re-write the associated communication and safety control modules in Java, in order to offer a wider users' field.

We equally plan to enhance the architecture of an intelligent virtual reality-based training system with as much as four software agents: training task-planning agent (now *Aker*), simulation agent, on-line instruction agent (now *Ash*), and an intelligent interface agent. The simulation agent will assist the student in predicting the robot path in order to follow

him with the moving camera. The intelligent interface agent will have as the main task assisting the distant-user in choosing the suitable (from the platform point of view) client-application. The *Ash* agent will be responsible for sending short text messages (SMS) to the "teacher" mobile phone when misuse of both software and equipments are "observed" outside the class timeframe.

We plan also to make agents more effective, i.e. they must be aware of each other. One way to do this is by exchanging messages. A second form of awareness involves the state of the agent's own environment, including characteristics of the computer on which it is executing and its network connection. A third involves self awareness: knowing its name, age, ontology, goals, areas of expertise and ignorance, and reasoning abilities. Finally, the agent should be aware of its physical environment (Huhns and Seshardri, 2000).

ACKNOWLEDGEMENTS

We gratefully acknowledge the support of Swiss National Science Foundation (Grant 7RUPJ062223, within the SCOPES framework)

REFERENCES

- Albu, Mihaela (2001). Virtual Laboratories and Remote Experimentation: a Realistic Perspective, Virtual Media Symposium, Bucharest, May 2001.
- Buiu, C. (1999). Artificial Intelligence in Education – State of the Art and Perspectives, *ZIFF Papiere*, 111, 1-105.
- Buiu, C., Mihaela Albu, I. Dumitrache, and D. Gillet (2001). Remote Automatic Control via Internet, in *Proceedings of the 13th International Conference on Control Systems and Computer Science*, CSCS13, Bucharest, 31 May-2 June 2001, pp. 76-83.
- Gillet D., H. A. Latchman, Ch. Salzmann, and O. D. Crisalle (2001). Hands-On Laboratory Experiments in Flexible and Distance Learning, *Journal of Engineering Education*, pp. 187-191, April 2001.
- Guru, A., P. Savory, and R. Williams. (2000). A web-based interface for storing and executing simulation models. In J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick (eds.) Department of Industrial and Management Systems Engineering, *Proceedings of the 2000 Winter Simulation Conference*, University of Nebraska, Lincoln, U.S.A.
- Huhns, M.N. and S. Seshardri (2000). Sensors + agents + networks = aware agents, in *IEEE Internet Computing*, Volume: 4 Issue: 3, pp. 84-86, South Carolina Univ., USA .

- Oeren, T. (2001) Impact Of Data On Simulation: From Early Practices To Federated And Agent-Directed Simulation, in A.W. Heemink, L. Dekker, H. de Swaan Arons, I. Smit and Th.L. van Stijn, Eds.: *Proceedings of EUROSIM 2001, Shaping Future with Simulation*, Delft, June 2001, NL
- Nakatsu, R. (1998). Towards the Creation of a New Medium for the Multimedia Era, *Proceedings of the IEEE*, Vol. 86, no. 5
- Salzmann Ch., D. Gillet, and P. Huguenin (2000) Introduction to Real-Time Control using LabVIEW™ with an Application to Distance Learning, *International Journal of Engineering Education "Special issue: LabVIEW Applications in Engineering Education"*, Vol. 16, No. 3, pp. 255-272.
- Talukdar, S. (1999). The next software revolution, in *Proceedings of the IEEE Power Engineering Society Summer Meeting*, 1999. Carnegie Mellon Univ., Pittsburgh, PA, USA, 843 - 845 vol.2