Synchronous[Q6] Collaboration of Virtual and Remote Laboratories

JARA CARLOS, CANDELAS FRANCISCO, TORRES FERNANDO,
DORMIDO SEBASTIÁN, ESQUEMBRE FRANCISCO

Carretera[Q2] *San Vicente del Raspeig, s/n, Alicante 03690, Spain*

Received 19 February 2009; accepted 16 July 2009

ABSTRACT: Virtual and remote laboratories (VRLs) are e-learning resources which enhance the accessibility of experimental setups providing a distance teaching framework which meets the student's hands-on learning needs. In addition, online collaborative communication represents a practical and a constructivist method to transmit the knowledge and experience from the teacher to students, overcoming physical distance and isolation. Thus, the integration of learning environments in the form of VRLs inside collaborative learning spaces is strongly desired. Considering these facts, the authors of this document present an original approach which enables user to share practical experiences while they work collaboratively through the Internet. This practical experimentation is based on VRLs, which have been integrated inside a synchronous collaborative e-learning framework. This article describes the main features of this system and its successful application for science and engineering subjects. © 2009 Wiley Periodicals, Inc. Comput Appl Eng Educ 17: 1–15, 2009; Published online in Wiley InterScience (www.interscience.wiley.com); DOI 10.1002/cae.20380

Keywords: collaborative learning; distance and remote teaching; learning environments

INTRODUCTION

Emerging technologies have allowed obtaining new ways of learning. Among them, it is worth pointing out the paradigm *Computer Supported Collaborative Learning* (CSCL), which was created for the use of technology as a mediation tool within collaborative methods of instruction [1]. CSCL systems have been integrating into web-based platforms giving new tools for distance teaching. These collaborative e-learning environments have caused a revolution in the academic community, since collaborative learning methods tend to encourage construction of knowledge, deeper understanding, and greater skill development [2].

There are two different types of CSCL environments according to the moment when the student–teacher interaction takes place: asynchronous and synchronous [3]. The first one allows data exchange in flexible time tables and remote access in an asynchronous way. However, this type of

Correspondence to J. Carlos (cajb@dfists.ua.es).

communication can cause feelings of isolation in the student and hence reduces his/her motivation [4]. Students do not receive instant feedback from their questions and cannot talk in real-time about results obtained in the learning activities. In contrast, synchronous environments enable the e-learning in a similar way to the traditional classroom, sharing experiences in real-time like face-to-face interaction. These systems usually use chats, audio, and video conferences [5], shared desktop [6], shared whiteboard [7], and shared 3D virtual worlds [8]. The integration of both CSCL systems in e-learning platforms is now having a significant impact on education outcomes.

Nowadays, CSCL synchronous environments are mainly focused in the learning of theoretical lessons. However, sometimes theory does not provide enough knowledge to students, especially in the field of engineering education. Practical experimentation is a vital part of science and engineering teaching that allows students to develop a deep understanding of theoretical lessons. To that end, virtual and remote laboratories (VRLs) are e-learning resources whereby students can learn through the Internet in a practical way and thus become aware of physical phenomena that are difficult to explain from just a theoretical point of view. VRLs represent distributed environments which are intended to perform both the simulation of a mathematical model of a real system proposed and remote experiments on this real equipment [9]. Their interactivity encourages students to play a more active role in the e-learning process and provides realistic hands-on experience [10]. Nevertheless, the majority of the VRLs added in web-learning environments are designed to be used individually, and they do not allow work group or collaboration among students and teachers [11]. The integration of VRLs inside collaborative learning environments can be seen in *eMersion* [12]. This web-based platform contains a series of VRLs whereby students can experiment and share results among other students or with teachers [13]. However, the collaboration in this CSCL environment is in an asynchronous way.

Taking into account the previous aspects, authors consider that several educational advantages could be achieved from combining VRLs with a synchronous collaborative learning, especially for education of practical experiences in technical or science subjects. Like in a real laboratory environment or traditional classroom, the new collaborative e-learning system presented in this article allows a group of students to share experiences at the same time they practice using VRLs. This system also permits teachers to track, supervise, and help students in their experimental exercises in a synchronous way. This can reduce strongly the teacher's tutorial process and allows students to resolve the practical exercises in a more guided way.

The system proposed combines the main advantages of VRLs and the synchronous collaborative learning practice. The VRLs are based in Java applets developed in *Easy Java Simulations* (EJS) [14], and the collaborative e-learning framework is a synchronized communication in real-time among them. Teachers and students will be enabled to share practical experiences in an e-learning synchronous environment by means of VRLs. In addition, the system can be applied in an easy way to all the applications (not only VRLs) developed in EJS and can be used as a main tool in any CSCL synchronous environment. This supposes a portable collaborative framework for distance teaching in several science disciplines, a feature which has not been implemented before in previous research projects.

The remainder of this article is organized as follows. Second Section discusses the related work which follows the synchronous approach. The following section describes basic aspects of EJS. In Fourth Section, the collaborative framework developed will be explained. Afterwards, a complete example about the generation and use of a collaborative class will be shown. Examples of collaborative distance sessions with VRLs are presented in Sixth Section. A reliable measure of the system's quality is described in Seventh Section. Some educational results are presented in Eight Section. Finally, some important conclusions are shown in Ninth Section.

RELATED WORK

Most collaborative tools which allow access to applications and collaboration among users have been addressed by screen-sharing systems [6]. However, these programs have several drawbacks such as their platform dependence, their low flexibility (only one collaborative session at the same time), and their need of a higher bandwidth. With the introduction of Java, it has been possible to overcome these problems. The use of this programming language has allowed the development of collaborative tools with a higher accessibility, portability, and platform independence.

Our approach represents a Java-based collaborative tool which permits to share VRLs embedded in Java applets in a transparent way. Since the appearance of Java language, many Web-and-Applet-based collaborative tools have emerged to improve CSCL environments, but none of which are able to share VRLs as explained in this article. Most of them require the use of an API, with the cost of modifying and implementing the source-code to make them collaborative. Among them, it is worth pointing out the following: the *Habanero* framework [15] is a programming tool that supports the development of real-time collaborative environments. *Java Collaborative Environment* (JCE) is an extended version of the Java-AWT, where mouse and keyboard events are intercepted and distributed among all the shared Java applets [16]. Finally, *Java Shared Data Toolkit* (JSDT) from Sun Microsystems adds collaboration features to Java applets. In contrast, the system presented in this article is not a new API and users do not have to change the source-code to create a collaborative environment with shared applets.

There are other systems which do not propose a new API for developing collaborative environments. This is the case of JETS [17], JAMM [18], JASMINE [19], and DECA [20]. JETS is a client-server framework which allows sharing Java applets. However, it can only share some specific multimedia applications. JAMM and JASMINE are systems which allow one to share almost any Java applet on the Internet. These client-server frameworks capture both AWT-based and Swing-based events from the user interaction and send them to all other participants connected to the server session. However, this approach cannot be applied to VRLs, since these learning objects often describe physical phenomena where some events and some updating of variables may happen without a user interaction. Finally, DECA offers a powerful tool to share all kind of applications such as Microsoft Office, CAD programs, etc., but not VRLs.

As mentioned before, there are software platforms that integrated VRLs inside collaborative environments. Most of them are based on asynchronous collaboration, where students can share experimental results and can communicate with tools such email or forums [12,21]. There are other approaches that include VRLs with real-time collaboration, where students can only visualize the experiments solved [22,23] and/or students can interact on the same application [24]. Nevertheless, the communication modules of these applications are specific according to their environments and they cannot be used for any other applications.

Our approach offers a new tool that allows any user to make collaborative any application developed in EJS. Moreover, the employment of this tool is very transparent, providing a powerful tool for e-learning and remote teaching. In contrast of screen-sharing programs, this system allows multiple sessions and it can be used with a low bandwidth connection. Its collaborative framework is based in peer-to-peer architecture (Fourth Section) which improves the client-server framework used in other systems. Besides, users only need a web browser and the *Java Runtime Environment* (JRE) installed in order to use this tool.

EASY JAVA SIMULATIONS: A POWERFUL TOOL FOR VIRTUAL AND REMOTE LABORATORIES

EJS is an open-source tool developed in Java, specifically designed for the creation of interactive dynamic simulations [14]. EJS was initially created for the Open Source Physics Project [25] which was established to construct and distribute curricular material for physics computation.

The most important reason for choosing EJS as the platform to apply our collaborative e-learning system is its simplicity of use. It is not necessary to have specific programming skills to develop an interactive simulation. EJS generates automatically the Java source code, the compiled files and the Java applet with the simulation embedded. This automatic code generation allows users to concentrate on describing the model simulation and they do not need a big investment of time to create a VRL.

The process to create an application with EJS includes the definition of the mathematical model and the view (i.e., the graphical interface). With regard to the model, the user must declare the variables that describe the system, must initialize them to initial values and must write the differential equations that establish how these variables change in time or under user interaction. With regard to the view, EJS provides a set of standard Java Swing, Java 2D/3D components to build it. These graphical components have certain properties that the user can connect with the model variables in order to set a link between the model and the view.

Finally, it can be considered that EJS is a powerful tool for developing VRLs. This software is being extensively used in the educational world and a lot of applications are being developed [26]. In addition, EJS' VRLs are being integrated both in CSCL environments [13] and networks of VRLs for teaching and learning collaboratively [27].

THE COLLABORATIVE FRAMEWORK

One of the most important advantages of using Java technology is that everybody can easily use it.

As mentioned before, it only needs a web browser and the JRE installed (any version from 1.4 to 1.6) to create a collaborative virtual class with the presented approach.

Components and Floor Control

A fundamental issue in a synchronous collaborative system is the floor control [28]. This term points out how the system's components share the computational resources. The main objective of the system developed is to offer a shared VRL that can be controlled in real-time by different members of a virtual class (students and teacher) and to be able to share the same experiments like in a traditional classroom. The shared VRL is composed of an applet for each class member. Therefore, there are two main kinds of components to coordinate: a teacher applet and some student applets. Initially, the teacher applet manages in real-time the virtual class and synchronizes all the student applets. He/she has a list of students connected in the virtual session and can disconnect any student at any moment. In order to have a suitable floor control, connected student applets are locked and they cannot interact with the shared VRL in a first moment. They only are able to see in real-time what the teacher is doing in the shared application. In this way, the collaborative session avoids collisions of events which can cause unwanted and incoherent results. One example of this problem could be that the real equipment of the VRL becomes uncontrollable because of unsuitable user interactions. Subsequently, in order to manage the virtual class like a traditional classroom, the system also has the feature of *chalk assignment*. With this option, the teacher gives permission to control the shared VRL to a specific student connected. The *chalk* only enables to manage the application, but not the virtual session.

Communication Framework

The communication framework is based in a server-less collaboration over TCP and UDP protocols. Student applets are connected directly with the teacher applet in a peer-to-peer (P2P) centralized overlay network. In contrast with server-based approaches [19], our e-learning system is focused in a server-less architecture. This communication method avoids the delays caused by the server processing in the data flow because the communication engine is embedded in the Java applets downloaded by the users. In addition, the number of network connections can be substantially decreased because the teacher applet can manage the session, the floor control, and the data exchange having higher control over the student applets.

As stated, the P2P network is centralized around the teacher applet. This last application is the central node of the collaborative class and contains a multi-thread communication module which manages the synchronization of all the applets that compose the shared VRL. Student applets are connected to the central node over TCP and UDP sockets performing a centralized network (Fig. 1).

Most collaboration tools synchronize applets using user event interactions. However, as commented before, this approach cannot be applied to applets which share a VRL. Authors have developed a method based in Java object tokens [28] which is able to synchronize in real-time all the applets connected to the virtual class. Java object tokens are small update messages which contain a *String* object that defines the action to be performed by other applets of the same session. The small amount of information sent optimizes network utilization and reduces the connection delay.

Since all the applets must be in the same state in any time, it is necessary to synchronize them. The communication framework developed provides a transport service suitable for all update data: a TCP-based channel for reliable messages and an UDP-based channel for fast messages. The TCP channel is used for update all the variables of the application because the transmission of the values needs the reliability of an ACK-based protocol. The UDP

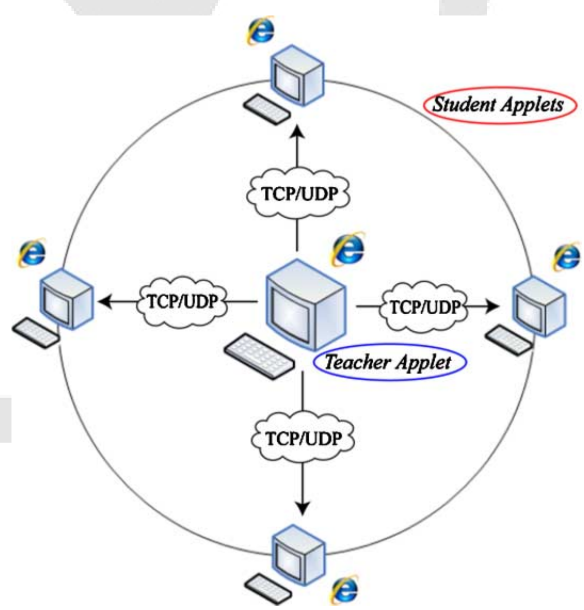

Figure 1 Centralized overlay network performed in the collaborative system.

channel is used to transmit the small changes in the user-interface and this requires to be quickly updated in the rest of the applets.

Software Architecture

The communication system explained in the previous subsection is embedded both the teacher and student applets. Therefore, a user (teacher or student) is able to share the VRL when the applet is downloaded with only a web browser that supports Java (1.4 o higher). This subsection describes both the teacher and student communication engine.

Teacher Architecture. Teacher applet is a multi-thread application that manages the synchronization among different student applets. Its software architecture is shown in Figure 2. At the beginning of the virtual session, a *Socket Server* object attends, in a fixed port (n), new TCP requests from the student applets. For each student connected, a new connection handler is added to the *Connection Array* with the IP student address, and a new reader handler is added to the *Reader Array* which manages feedback messages from students. To synchronize the applets, the object *Sender* retransmits update messages to all the student applets connected. As commented previously, these commands are embedded in Java object tokens and sent through the TCP and UDP channels established. Each token is a Java object composed by a String label which identifies the message command and a String data object with the variable values to update.

When a student has the *chalk* control, the teacher applet behaves like a student applet. In this situation, the controls of the teacher applet are locked and it receives orders from the student applet which has the *chalk* assigned. The teacher communication system activates the corresponding reader thread from *Reader Array* to receive orders through the TCP and UDP channels. Because the virtual class uses a centralized approach, only the teacher applet receives update messages from the student applet with the *chalk* assigned. This way, the communication engine of the teacher applet has to retransmit these orders to the rest of the student applets.

Student Architecture. The software architecture of a student applet is shown in Figure 3. The object *Receiver* is a thread which listens to update messages from the teacher through the TCP and UDP channels. Finally, the object *Sender* is used to send update messages when the *chalk* is assigned to the student applet.

Latecomers

When the teacher applet begins the collaborative virtual class, it supposes that all student applets are connected. However, a student can join in a session already in progress. To synchronize this newcomer with the rest of the applets, the teacher applet sends through the TCP connection the current state of the shared VRL. This update message contains the current value of all the variables of the application. Therefore, the latecomer is ready to receive update messages from the teacher applet and to take the collaborative virtual class in a synchronous way.

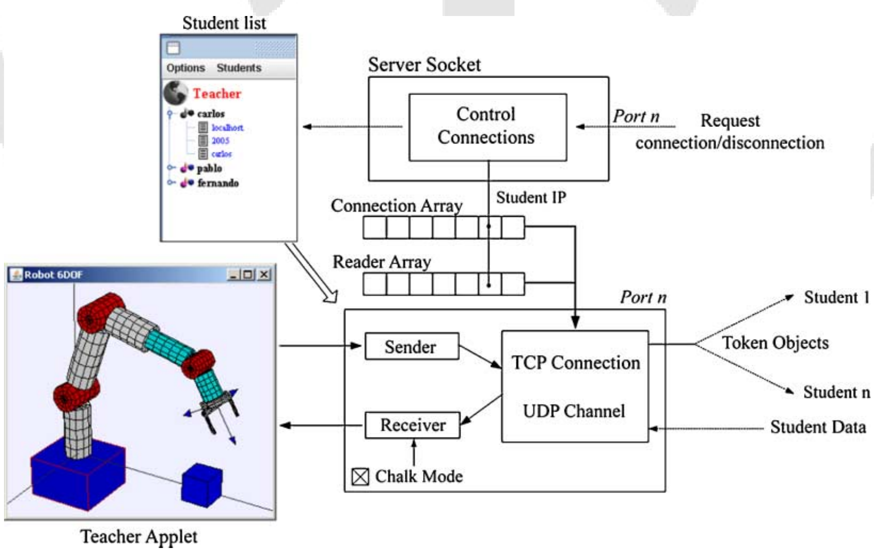

Figure 2 Multi-thread teacher architecture.

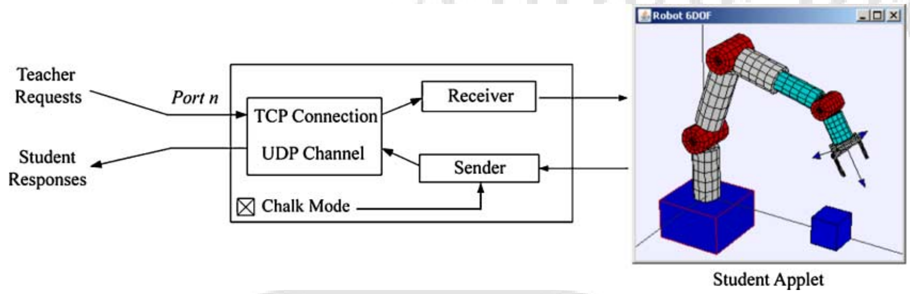

Figure 3 Multi-thread student architecture.

CREATION OF A COLLABORATIVE SESSION

This section describes how to create an e-learning collaborative session by means of the approach presented here. This process is comprised of two points:

1. The generation of a Java applet which is ready to teach in a virtual class (i.e., the teacher applet).
2. The generation of an URL address for students in a public web server, where they can download their collaborative applets.

This section will be explained using a heat-flow process simulation as example [29]. Of course, this application is only a simple example of the power of EJS.

Teacher Applet Generation

As already commented, EJS creates an HTML page with the simulation embedded in Java applet form. Therefore, after developing the application (model and view) in EJS, the user only has to complete two fields in the EJS options to generate a collaborative-enabled teacher applet:

1. *URL server*: URL address of a public web server where the Java applets will be installed.
2. *Port*: Port number which will be opened for TCP and UDP communications in the virtual class. The same port is used for both protocols, not only in teacher applet, but also in student applets. This simplifies the configuration of network devices and firewalls (Fig. 4).

After filling these fields, a teacher applet prepared for collaboration is created. The communication engine is embedded in the same compiled classes that EJS generates. Figure 5 shows the appearance of the simulation proposed, where the JavaScript control called ''Collaborative Mode'' provide access to the functionality needed to generate the student applets and to start the collaborative class.

Dynamic Student Applet Generation

In order to provide an URL address to the students in a virtual class, authors have developed a method to generate dynamically an HTML page in a public web server. This process is completely necessary to automatically provide the teacher's public IP address and the teacher's opened port for student applets in a transparent way.

The dynamic generation module is a PHP page installed in a public web server. Its functions are to get the teacher's public IP and to create an HTML file ready to be accessed by students. The student URL generation is performed directly from the teacher applet. The ''Collaborative Mode'' control executes a dialog called *HTML Generation* where the following fields appear: *IP address*, where the teacher has to choose between local IP (for Intranet collaboration) or public IP (for Internet collaboration); *Identification and Password*, required for security; and *Server Site* and *Port*, which are the previous fields filled in EJS options. After filling them, the communication system of the teacher applet sends the dialog data via the

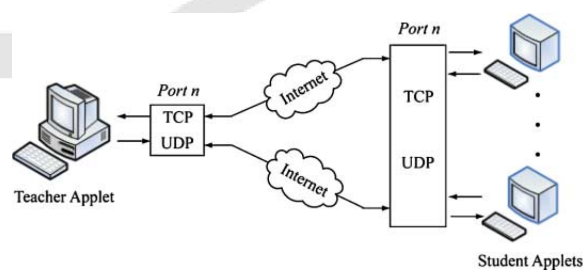

Figure 4 Port configuration of the collaborative framework.

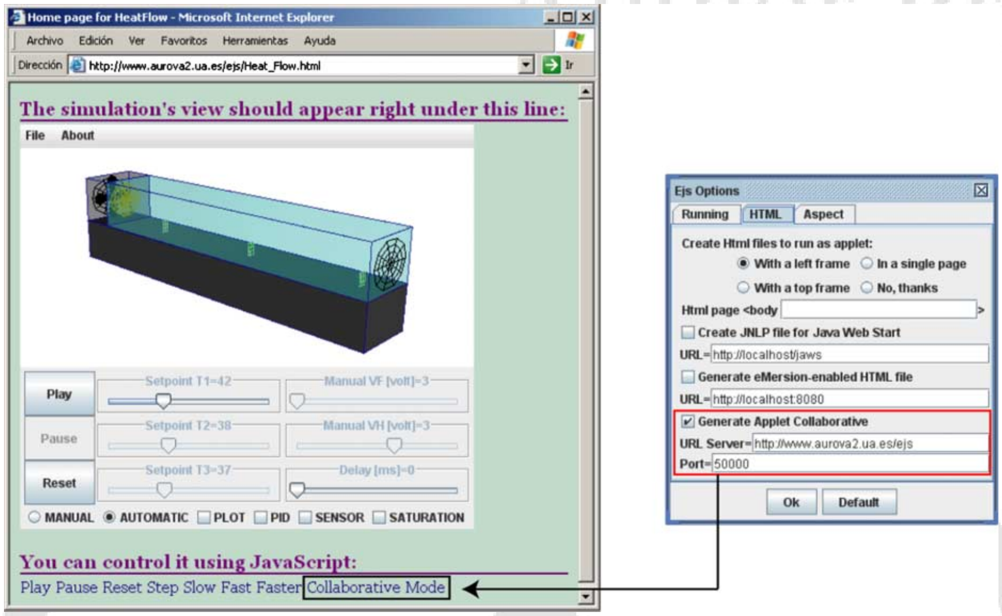

Figure 5 Generation of a teacher applet with a heat-flow simulation embedded.

HTTP protocol. The PHP module located in the public web server receives this object data and generates an HTML file with the student applet embedded, ready to be accessed by students. The teacher's public IP and teacher's port are given, as parameters, to the student applet from the HTML generated (Fig. 6). The field *Applet Student Generated* will show the URL to be used by students to access to their collaborative applets. The teacher only has to send it to students by means of a simple collaborative auxiliary tool such as e-mail, forums, or chat. If the previous process is successful, the control "Connect the Server" will be enabled in the *HTML Generation* dialog. After activating this control, the teacher applet will be listening to student applet requests on the port opened.

After that, students can connect to a collaborative virtual environment with only one URL address. They have to connect to the web address provided, activate the control "Virtual Class" and introduce a name in the dialog (Fig. 7). This is the identification for the teacher's student list.

CASES OF STUDY

In this section, the synchronization results of the collaborative system developed will be shown by means of two experimental examples. Both experiments were performed using the synchronous approach together with the free software *Skype* (on-line chat, audio and video conferencing), which was

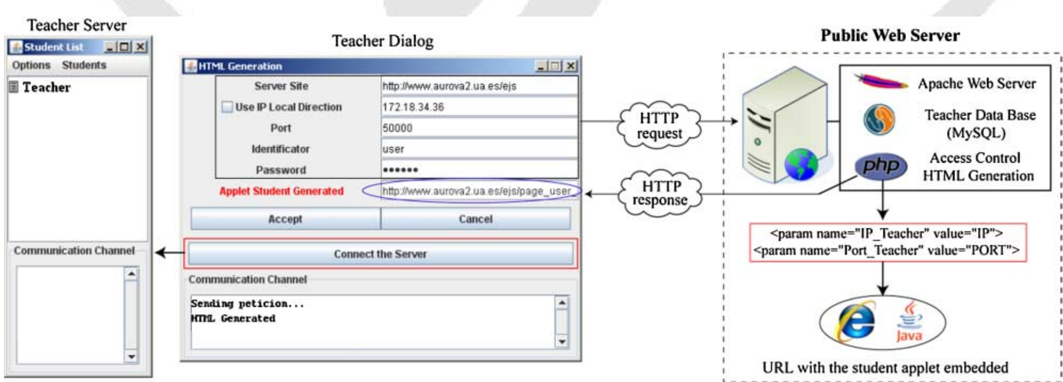

Figure 6 Dynamic generation of a student URL address.

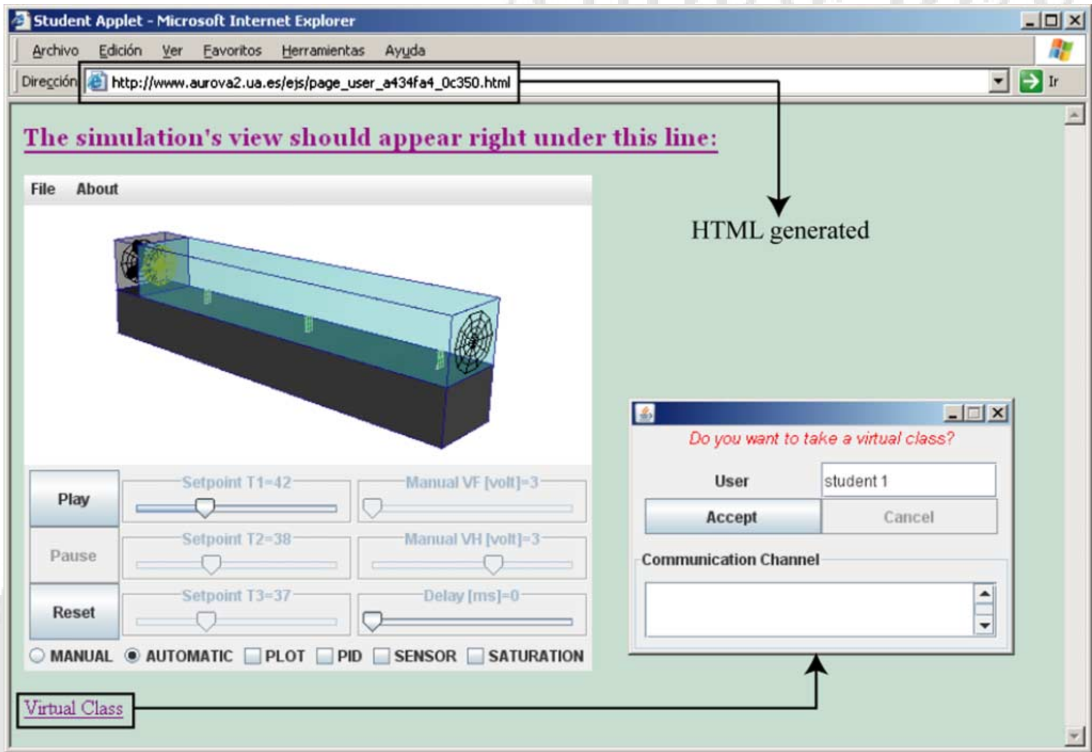

Figure 7 Student applet generated.

employed as complementary tool to communicate teacher and students. Users can utilize any software tool for chat and video conferencing (Messenger, Globus Toolkit, Java Multimedia Tools, etc.) since its use is independent of the collaborative system developed. In both examples, the synchronized applets are in the same model state. They show that the view and model of the applets connected in the virtual class have the same variable values at the same time.

Collaboration in a Virtual Laboratory: A Three Tank System

The first example considered is a virtual laboratory which simulates a three-tank process. This synchronous virtual laboratory has been created by just building an existent application [30] with the synchronous collaborative framework for EJS described in this manuscript. The experiment proposed consists of a virtual class to teach the effects of a Proportional-Integral-Derivative (PID) controller in the system dynamics.

From the moment of the connection by the students in the collaborative session, their interfaces are disconnected. Student cannot experiment with the shared application and the teacher applet manages in real-time the simulation evolution. Figure 8 shows this case, where the teacher is modifying the parameters of the PID controls in his/her applet (upper window) and a student applet (lower window) is visualizing these changes in a virtual session with three students connected. The teacher applet has the session control in order to explain practical concepts about PID control effects by means of the interaction in the shared application. The student applets are lock (controls disabled) and they are learning within the collaborative system from teacher's comments and interaction. With regard to the synchronization, not only all the variables of the shared application are synchronized (emphasized fields in the Fig. 8), but also the graphic representations show the same evolution.

After finishing teacher explanation, the *chalk* control is assigned to a student in order to enable him/her to experiment with the shared application. This way, the student can practice PID control concepts acquired during the virtual class. Here, the teacher applet is lock and the student is who has the control of the shared application (Fig. 9). The rest of students connected visualize in real-time that this student with is doing the virtual laboratory.

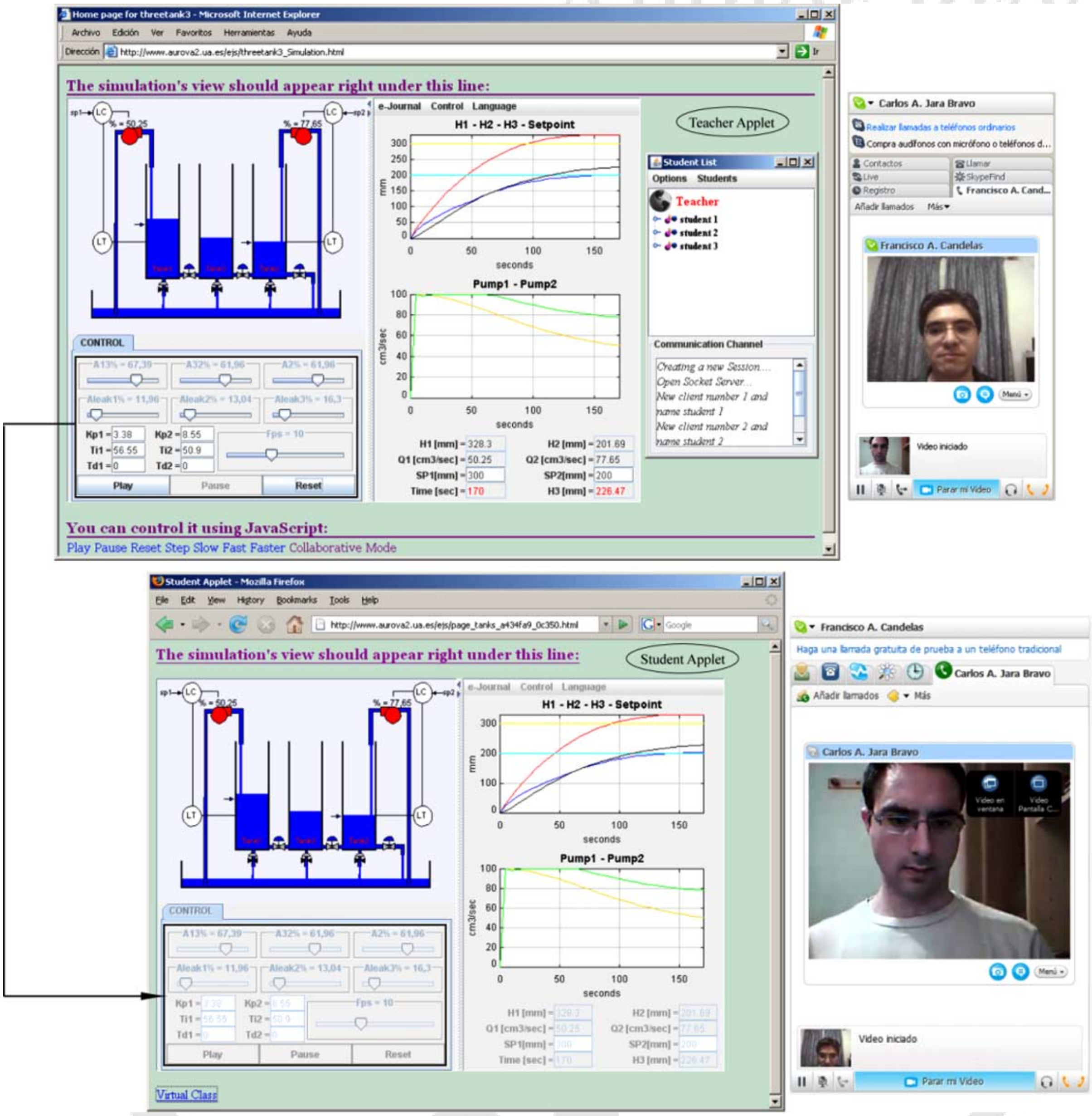

Figure 8 Teacher control in a virtual class with a three-tank system simulation.

Collaboration in a Virtual and Remote Laboratory: A Telerobotic System

The second example is a VRL for a telerobotic system which allows user to simulate and test positioning commands for a robot by means of a virtual environment, as well as execute high level commands in a real remote robot through the Internet. This collaborative VRL has been obtained from the one described in Ref. [31] by building it with the synchronous collaborative version of EJS. By means of a VRL like this and the synchronous approach presented here, the teacher can explain to students complex concepts about Robotics and teach the functionality of the telerobotic system. The experiment proposed consists of simulating a trajectory in the virtual environment and teleoperating it in the real robotic plant.

Figure 10 shows the teacher applet and a student applet for a collaborative session with this VRL where the 3D environment and the graphical representation are synchronized. Here, the teacher is controlling the virtual environment in order to teach concepts about path planning algorithms.

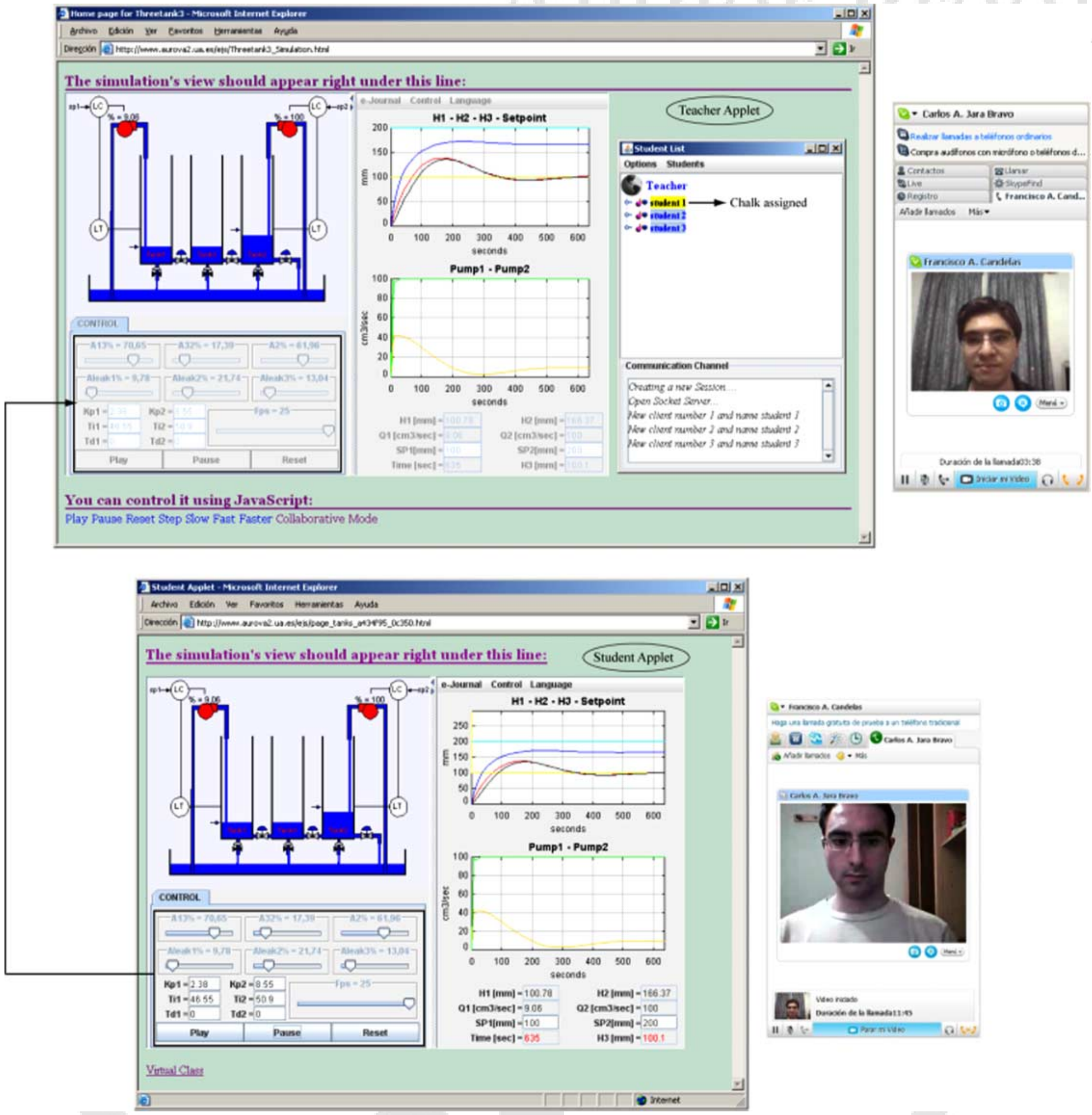

Figure 9 Student control in a virtual class with a three-tank system simulation.

In addition, the students connected can visualize the teleoperation process by means of a window which displays the real plant (Figs. 10 and 11). After simulating the robot trajectory, this is teleoperated in the real robotic plant (Fig. 11).

QUALITY SYSTEM OVERVIEW

The communication engine developed must be able to support collaborative real-time applications within the acceptable parameters of quality of service for users. This section presents an evaluation of the delays obtained during a virtual class performed through the Internet.

The teacher applet sends to student applets small update messages through the UDP channel. A parameter that measures the quality of our system is the average time from a UDP message request to all the UDP messages responses. This parameter will be called Collaborative Time (CT). It includes transport, networking, and physical layer delays among teacher and student applets.

Internet performance evaluation was carried out from different places with common Internet

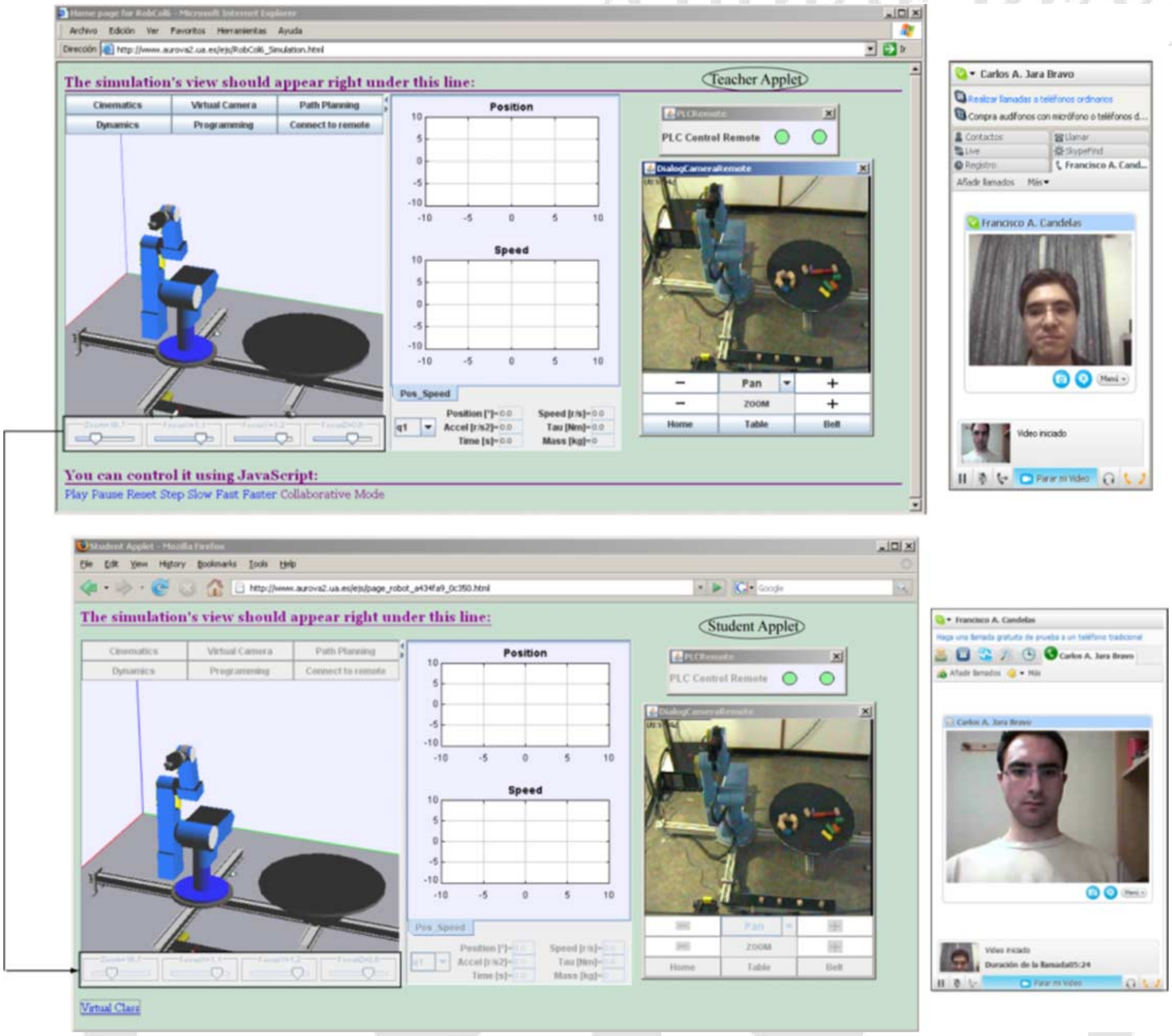

Figure 10 Virtual class with a virtual and remote telerobotic laboratory.

connections in the same country. Internet access ranged from a 1 Mbps/320 Kbps ADSL to a 5 Mbps/300 Kbps cable-modem. The system was evaluated with two, four, and six students connected, in addition to the necessary teacher. Figure 12a shows the average and deviation CT values measured in the Internet virtual session.

As expected, the CT increases with the number of student applets connected. Analyzing the results, the small time delay introduced in the three experiments does not produce any sense of blocking and the synchronous system works correctly in terms of interaction and perception.

Figure 12b shows the CT delay variation during 1 min in the virtual class. As already mentioned in The Collaborative Framework Section, UDP protocol offers an unreliable stream, which is affected by the Internet traffic, and some UDP packets can be lost. When a UDP packet is lost, teacher applet evolution waits for student responses by a configured timeout, in this case 150 ms (Fig. 12b). After that, teacher applet sends an update of all the variables through TCP channel. Thus, the system evolution of the shared applications is only blocked by the configured timer expiration.

EDUCATIONAL EVALUATION

At the moment, the collaborative system is being used in the courses *Computer Process Control and Robots and Sensorial Systems* in the Computer Science Engineering degree at the University of Alicante (some of the applications used by authors

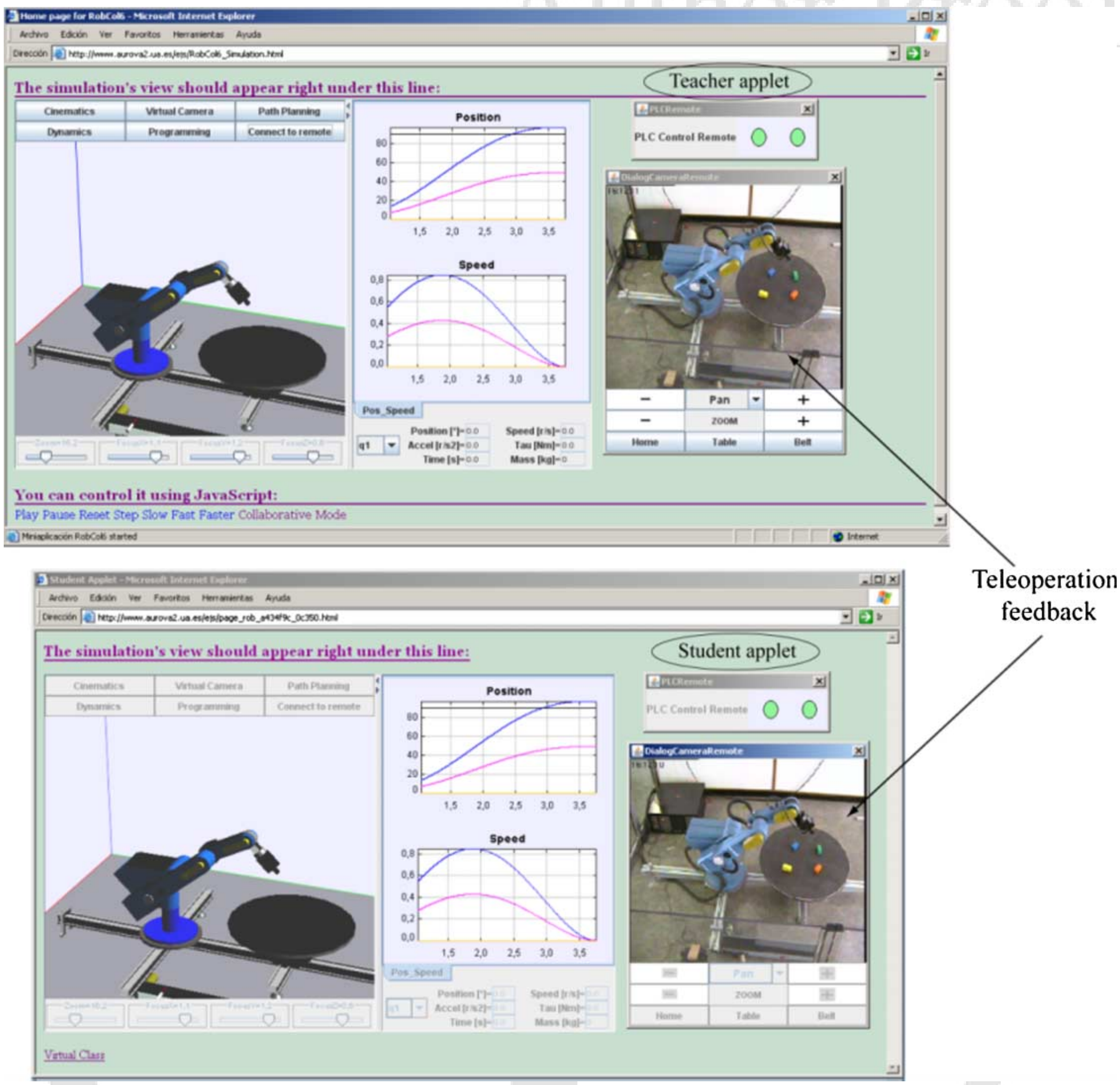

Figure 11 Collaborative teleoperation of a joint trajectory.

in these courses are available from the web http://www.aurova2.ua.es). The educational methodology is based on two points:

1. *Practical lessons at the University*: The collaborative synchronous system is used in a PC room with a LAN connection. Teacher explains process control and robotic topics by means of simulations or VRLs and students can follow the class on their own screen. In case of student doubts, the *chalk* mode is used. Thus, the student is able to manipulate the shared application and to explain his/her doubt.
2. *Practical lessons through the Internet*: Teachers are using this collaborative system together with *Skype* to resolve student doubts from home. During the teacher tutorial time, students connect to the teacher and ask questions through *Skype*. Teachers, by means of some VRL and the collaborative synchronous system, resolve the student doubts.

In order to have student feedback, a study was carried out to verify the usefulness and efficiency of the collaborative system presented. Questionnaire items were focused on three issues:

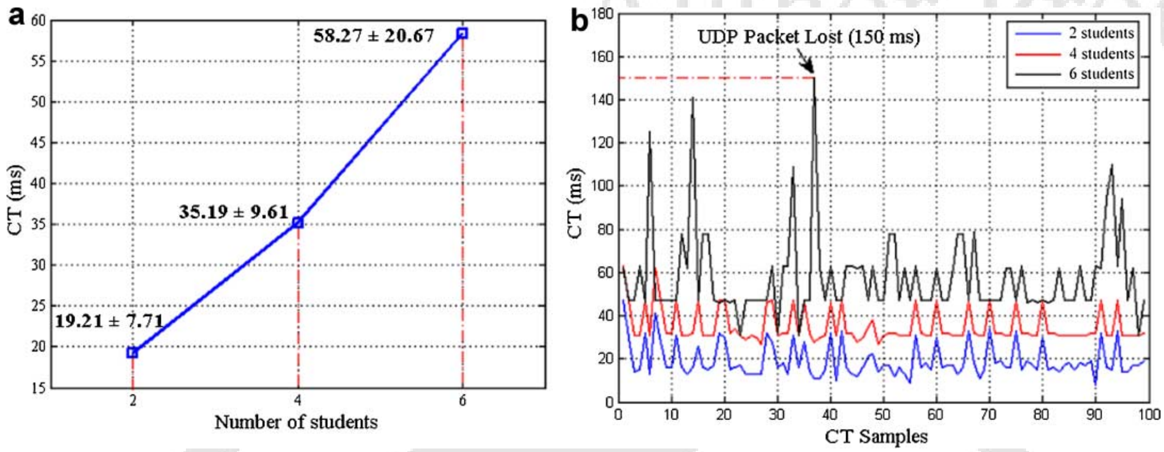

Figure 12 (a) Average and deviation CT values; (b) CT delay variation in the virtual class.

1. The suitability of VRL in the learning of relevant control and robotic concepts (I1).
2. The functionality of the collaborative system (I2).
3. The effectiveness of the synchronous collaboration in the learning process (I3).

Table 1 shows the questionnaire made to 25 students from the courses. Responses were rated on a five point Likert scale ranking from strongly agree (1) to strongly disagree (5). The statistical results obtained are reported in Table 2. Analyzing them, more than half of the students think that VRLs together with the synchronous collaborative system proposed are useful for distance practical lessons. In addition, most of the students consider that the VRL is an efficient tool in the learning process which encourages their ability to understand control and robotic concepts. They also consider efficient the use of a popular VoIP tools (video and audio conference) such as *Skype* in the collaborative environment to obtain a suitable teacher's feedback.

In relation to the software to be installed, none of the students had problems with the installation of the necessary JRE. It should be emphasized that our students study Computer Science and they do not use to have problems regarding to new technologies issues. They adapt quickly to the user interfaces of the VRLs.

With regard to the educational results obtained, authors compared students who used a traditional system where they practice without collaboration (academic year 2006−2007), with the students who used the new proposed system (academic year 2007−2008), solving the same experiments. All the students passed the course. However, there is a few number of students who got better marks with the use of the collaboration system: 13% more with A qualification and 9% more with B qualification. The global qualification between 2 years increased from 7.39 to 7.82. In addition, for the first year was needed a long tutorial process since it was used a traditional communication (emails and forums). In the second year, students learned the main concepts more quickly than the first year thanks to real-time

Table 1 Questionnaire to Obtain User Feedback

Questions	Issue
In general, do you find easy the use of VRLs for practical lessons?	I1
Did the VRL help you to understand theoretical concepts?	I1
Did you find it easy to do the practical exercises with the VRLs?	I1
Did you find easy the use of the collaborative system?	I2
Did the collaborative system work correctly? Write the problems found	I2
Was the synchronization good in terms of perception and interaction?	I2
Was the synchronous collaboration among the VRLs useful?	I3
Did you learn from the teacher's real-time feedback?	I3
Was the collaboration useful to share knowledge with your classmates?	I3

Table 2 Student Questionnaire Results in Percentage of Agreement Per Issue

Issue	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
I1	64	24	8	4	0
I2	52	24	10	10	4
I3	48	24	20	8	0

sessions. In this way, students were able to resolve the experiments proposed before. Some students also were grateful because the synchronous system allowed them to resolve the experiments in a more guided way.

In addition, the collaborative system is going to be used among different universities in a network of VRLs [27]. The universities involved have important research activities in this field of and they have been creating VRLs with EJS for many years [13,29]. They are getting ready for the next coming *European Higher Education Area* [32], in which the e-learning and the team work will occupy very important roles.

CONCLUSIONS

In this article, a new web-learning system which combines two outstanding educational resources, the VRLs and the synchronous collaborative learning practice, has been presented. This approach has achieved a new method to share knowledge in a synchronous way based in experiences over VRLs.

This e-learning system represents a portable collaboration framework for interactive applications developed in EJS. The approach has been integrated in a new software version and can be applied in a transparent way to all the VRLs developed in this platform. In addition, EJS makes easy the task of creating a VRL and provides a large amount of interactive applications developed.

Nowadays, many existing collaborative tools use screen-sharing applications to perform synchronous communication in remote experimentation. In comparison with these tools, a system like the presented here provides an easy way to take a virtual class without using any complex external application. It only needs a standard web browser that supports Java. In addition, this system can be used as a main tool for CSCL synchronous environments for teaching in any education discipline.

The results obtained in the network evaluation were successful. The delay measured in a virtual class of six students with a common Internet connection was lower than 150 ms in the worst case. Therefore, the collaborative synchronous framework developed is inside of the acceptable parameters of quality of service.

Finally, authors are currently working on applying the collaborative synchronous system presented to remote network laboratories.

ACKNOWLEDGMENTS

The work presented in this article is supported by the ‘‘Ministerio de Educación y Ciencia’’ of the Spanish Government through FPI grants program and research project DPI2005-06222. Authors would like to thank for this financial support.

REFERENCES

[1] K. Koschmann, Toward a theory of computer support for collaborative learning, J Lear Sci 3 (1994), 219–225.
[2] M. Alavi, Computer-mediated collaborative learning: An empirical evaluation, MIS Q 18 (1994), 159–174.
[3] G. Bafoustou and G. Mentzas, Review[Q3] and functional classification of collaborative systems, Int J Inf Manage 22 (2002), 281–305.
[4] M. Kamel, A. Taylor, and A. Breton, A synchronous communication experiment within an on-line distance learning program: A case study, Telemed J e-Health 11 (2005), 583–593.
[5] P. Isenhour, D. Carroll, D. Neale, M. Rosson, and D. Dunlap, The virtual school: An integrated collaborative environment for the classroom, J Educ Technol Soc 3 (2000), 74–86.
[6] ‘‘Microsoft netmeeting,’’ http://www.microsoft.com, 2000.
[7] Z. Yang and Q. Liu, Research and development of web-based virtual online classroom, Comput Educ 48 (2007), 171–184.
[8] M. Teresa, M. Gavin, and B. Michela, Virtual reality for collaborative e-learning, Comput Educ 50 (2008), 1339–1353.
[9] S. Dormido, Control learning: Present and future, Annu Rev Control 28 (2004), 115–136.
[10] S. Dormido, R. Dormido, J. Sanchez, and N. Duro, The role of interactivity in control learning, Int J Eng Educ 21 (2005), 1122–1133.
[11] L. Moreno, C. Gonzalez, I. Castilla, E. Gonzalez, and J. Sigut, Applying a constructivist and collaborative methodological approach in engineering education, Comput Educ 49 (2007), 891–915.
[12] D. Gillet, Collaborative web-based experimentation in flexible engineering education, IEEE Trans Educ 48 (2005), 696–704.
[13] R. Dormido, H. Vargas, N. Duro, J. Sanchez, S. Dormido-Canto, G. Farias, F. Esquembre, and S.

Dormido, Development of a web-based control laboratory for automation technicians: The three-tank system, IEEE Trans Educ 51 (2008), 35–44.
[14] F. Esquembre, Easy java simulations: A software tool to create scientific simulations in java, Comput Phys Commun 156 (2004), 199–204.
[15] A. Chabert, E. Grossman, L. Jackson, S. Pietrowiz, and C. Seguin, Java object-sharing in habanero, Commun ACM 41 (1998), 69–76.
[16] H. Abdel-Wahab, B. Kvande, O. Kim, and J. P. Favreau, "An internet collaborative environment for sharing java applications," Proceedings of the 6th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS '97), IEEE Computer Society, 1997.
[17] S. Shirmohammadi, J. C. De Oliveira, and N. D. Georganas, Applet-based telecollaboration: A network-centric approach, IEEE Multimedia 5 (1998), 64–73.
[18] J. Begole, C. Struble, C. Shaffer, and R. Smith, "Transparent sharing of java applets: A replicated approach," Proceedings of the 10th annual ACM symposium on User interface software and technology, ACM, Canada, 1997.
[19] S. Shirmohammadi, A. E. Saddik, N. Georganas, and R. Steinmetz, Jasmine: A java tool for multimedia collaboration on the internet, Multimedia Tools Appl 19 (2003), 5–19.
[20] L. Petropoulakis and F. Flood, Design and development of a general purpose collaborative environment, Int J Appl Technol 29 (2007), 2–10.
[21] W. Min, S. Jin-Hua, Z. Gui-Xiu, and Y. Ohyama, Internet-based teaching and experiment system for control engineering course, IEEE Trans Ind Electron 55 (2008), 2386–2396.
[22] W. YuLung, C. TeYi, J. BinShyan, L. TsongWuu, and L. YaoHui, A web-based dual mode virtual laboratory supporting cooperative learning, 18th International Conference on Advanced Information Networking and Applications (AINA), 2004, 642–647.
[23] S. Elio, M. Sergio, G. Rosario, M. Catalina, L. Eugenio, O. Nuria, M. Francisco, D. Gabriel, and C. Manuel, Development and interaction between LMS services and remote labs, Int J Online Eng 4 (2008), 35–41.
[24] J. Machotka and Z. Nedic, "From the collaborative environment of remote laboratories to the global collaboration," Proceedings of the 5th Remote Engineering & Virtual Instrumentation (REV 2008), Germany, 2008.
[25] W. Christian and M. Belloni, Developing open source programs for science and mathematics, Proceedings of the International Conference on Computer as a Tool (EUROCON), 2003, 15–19.
[26] "Easy java simulations' wiki," http://fem.um.es/EjsWiki, 2007.
[27] "Automatl@bs project," http://lab.dia.uned.es/automatlab/index_en.html, 2008.
[28] H. Dommel and J. Garcia, Floor control for multimedia conferencing and collaboration, Multimedia Syst 5 (1997), 23–38.
[29] H. Vargas, J. Sanchez, G. Farias, S. Dormido, R. Dormido, S. Canto, and F. Esquembre, "Web-based learning resources for vocational training for automation technicians," Proceedings of the IFAC Symposium on Advances Control Education, Spain, 2006.
[30] R. Dormido, H. Vargas, N. Duro, S. Sanchez, S. Dormido, G. Farias, and F. Esquembre, Development[Q4] of a web-based control laboratory for automation technicians: The three-tank system, IEEE Trans Educ 51, 33–44.
[31] C. A. Jara, F. A. Candelas, and F. Torres, An advanced interactive interface for robotics e-learning, Int J Online Eng 4 (2009), 17–25.
[32] "European[Q5] higher education area," http://en.wikipedia.org/wiki/European_Higher_Education_Area, 2000.

Q1: Please check the suitability of the short title on the odd-numbered pages. It has been formatted to fit the journal's 45-character (including spaces) limit.
Q2: Please check the author names, affiliation, and correspondence address.
Q3: Please check the page range.
Q4: Please provide the year.
Q5: Please provide author biographies and photos.
Q6: Author: Color figures will appear in black and white in the print publication unless you are willing to assume the cost. Please check.

111 RIVER STREET, HOBOKEN, NJ 07030

ELECTRONIC PROOF CHECKLIST, COMPUTER APPLICATIONS IN ENGINEERING EDUCATION

*****IMMEDIATE RESPONSE REQUIRED*****

Please follow these instructions to avoid delay of publication.

☐ **READ PROOFS CAREFULLY**

- This will be your only chance to review these proofs.
- Please note that the volume and page numbers shown on the proofs are for position only.

☐ **ANSWER ALL QUERIES ON PROOFS** (Queries for you to answer are attached as the last page of your proof.)

- Mark all corrections directly on the proofs. Note that excessive author alterations may ultimately result in delay of publication and extra costs may be charged to you.

☐ **CHECK FIGURES AND TABLES CAREFULLY** (Color figures will be sent under separate cover.)

- Check size, numbering, and orientation of figures.
- All images in the PDF are downsampled (reduced to lower resolution and file size) to facilitate Internet delivery. These images will appear at higher resolution and sharpness in the printed article.
- Review figure legends to ensure that they are complete.
- Check all tables. Review layout, title, and footnotes.

☐ **COMPLETE REPRINT ORDER FORM**

- Fill out the attached reprint order form. It is important to return the form even if you are not ordering reprints. You may, if you wish, pay for the reprints with a credit card. Reprints will be mailed only after your article appears in print. This is the most opportune time to order reprints. If you wait until after your article comes off press, the reprints will be considerably more expensive.

☐ **ADDITIONAL COPIES**

- If you wish to purchase additional copies of the journal in which your article appears, please contact Neil Adams at (201) 748-8839, fax (201) 748-6021, or E-mail at nadams@wiley.com

RETURN ☐**PROOFS**
☐**REPRINT ORDER FORM**
☐**CTA (If you have not already signed one)**

RETURN WITHIN 48 HOURS OF RECEIPT VIA E-MAIL OR FAX TO 201-748-6182

QUESTIONS?

Laura Espinet, Associate Production Manager
Phone: 201-748-8884
E-mail:lespinet@wiley.com
Refer to journal acronym and article production number
(i.e., CAE 00-001 for *Computer Applications in Engineering Education* ms 00-001).

COPYRIGHT TRANSFER AGREEMENT

WILEY-BLACKWELL

Date: ____________________ Contributor name: ____________________

Contributor address: ____________________

Manuscript number (Editorial office only): ____________________

Re: Manuscript entitled ____________________

____________________ (the "Contribution")

for publication in ____________________ (the "Journal")

published by ____________________ ("Wiley-Blackwell").

Dear Contributor(s):
Thank you for submitting your Contribution for publication. In order to expedite the editing and publishing process and enable Wiley-Blackwell to disseminate your Contribution to the fullest extent, we need to have this Copyright Transfer Agreement signed and returned as directed in the Journal's instructions for authors as soon as possible. If the Contribution is not accepted for publication, or if the Contribution is subsequently rejected, this Agreement shall be null and void. **Publication cannot proceed without a signed copy of this Agreement.**

A. COPYRIGHT

1. The Contributor assigns to Wiley-Blackwell, during the full term of copyright and any extensions or renewals, all copyright in and to the Contribution, and all rights therein, including but not limited to the right to publish, republish, transmit, sell, distribute and otherwise use the Contribution in whole or in part in electronic and print editions of the Journal and in derivative works throughout the world, in all languages and in all media of expression now known or later developed, and to license or permit others to do so.

2. Reproduction, posting, transmission or other distribution or use of the final Contribution in whole or in part in any medium by the Contributor as permitted by this Agreement requires a citation to the Journal and an appropriate credit to Wiley-Blackwell as Publisher, and/or the Society if applicable, suitable in form and content as follows: (Title of Article, Author, Journal Title and Volume/Issue, Copyright © [year], copyright owner as specified in the Journal). Links to the final article on Wiley-Blackwell's website are encouraged where appropriate.

B. RETAINED RIGHTS

Notwithstanding the above, the Contributor or, if applicable, the Contributor's Employer, retains all proprietary rights other than copyright, such as patent rights, in any process, procedure or article of manufacture described in the Contribution.

C. PERMITTED USES BY CONTRIBUTOR

1. Submitted Version. Wiley-Blackwell licenses back the following rights to the Contributor in the version of the Contribution as originally submitted for publication:

a. After publication of the final article, the right to self-archive on the Contributor's personal website or in the Contributor's institution's/employer's institutional repository or archive. This right extends to both intranets and the Internet. The Contributor may not update the submission version or replace it with the published Contribution. The version posted must contain a legend as follows: This is the pre-peer reviewed version of the following article: FULL CITE, which has been published in final form at [Link to final article].

b. The right to transmit, print and share copies with colleagues.

2. Accepted Version. Re-use of the accepted and peer-reviewed (but not final) version of the Contribution shall be by separate agreement with Wiley-Blackwell. Wiley-Blackwell has agreements with certain funding agencies governing reuse of this version. The details of those relationships, and other offerings allowing open web use, are set forth at the following website: http://www.wiley.com/go/funderstatement. NIH grantees should check the box at the bottom of this document.

3. Final Published Version. Wiley-Blackwell hereby licenses back to the Contributor the following rights with respect to the final published version of the Contribution:

a. Copies for colleagues. The personal right of the Contributor only to send or transmit individual copies of the final published version in any format to colleagues upon their specific request provided no fee is charged, and further-provided that there is no systematic distribution of the Contribution, e.g. posting on a listserve, website or automated delivery.

b. Re-use in other publications. The right to re-use the final Contribution or parts thereof for any publication authored or edited by the Contributor (excluding journal articles) where such re-used material constitutes less than half of the total material in such publication. In such case, any modifications should be accurately noted.

c. Teaching duties. The right to include the Contribution in teaching or training duties at the Contributor's institution/place of employment including in course packs, e-reserves, presentation at professional conferences, in-house training, or distance learning. The Contribution may not be used in seminars outside of normal teaching obligations (e.g. commercial seminars). Electronic posting of the final published version in connection with teaching/training at the Contributor's institution/place of employment is permitted subject to the implementation of reasonable access control mechanisms, such as user name and password. Posting the final published version on the open Internet is not permitted.

d. Oral presentations. The right to make oral presentations based on the Contribution.

4. Article Abstracts, Figures, Tables, Data Sets, Artwork and Selected Text (up to 250 words).

a. Contributors may re-use unmodified abstracts for any non-commercial purpose. For on-line uses of the abstracts, Wiley-Blackwell encourages but does not require linking back to the final published versions.

b. Contributors may re-use figures, tables, data sets, artwork, and selected text up to 250 words from their Contributions, provided the following conditions are met:

(i) Full and accurate credit must be given to the Contribution.
(ii) Modifications to the figures, tables and data must be noted. Otherwise, no changes may be made.
(iii) The reuse may not be made for direct commercial purposes, or for financial consideration to the Contributor.
(iv) Nothing herein shall permit dual publication in violation of journal ethical practices.

D. CONTRIBUTIONS OWNED BY EMPLOYER

1. If the Contribution was written by the Contributor in the course of the Contributor's employment (as a "work-made-for-hire" in the course of employment), the Contribution is owned by the company/employer which must sign this Agreement (in addition to the Contributor's signature) in the space provided below. In such case, the company/employer hereby assigns to Wiley-Blackwell, during the full term of copyright, all copyright in and to the Contribution for the full term of copyright throughout the world as specified in paragraph A above.

2. In addition to the rights specified as retained in paragraph B above and the rights granted back to the Contributor pursuant to paragraph C above, Wiley-Blackwell hereby grants back, without charge, to such company/employer, its subsidiaries and divisions, the right to make copies of and distribute the final published Contribution internally in print format or electronically on the Company's internal network. Copies so used may not be resold or distributed externally. However the company/employer may include information and text from the Contribution as part of an information package included with software or other products offered for sale or license or included in patent applications. Posting of the final published Contribution by the institution on a public access website may only be done with Wiley-Blackwell's written permission, and payment of any applicable fee(s). Also, upon payment of Wiley-Blackwell's reprint fee, the institution may distribute print copies of the published Contribution externally.

E. GOVERNMENT CONTRACTS

In the case of a Contribution prepared under U.S. Government contract or grant, the U.S. Government may reproduce, without charge, all or portions of the Contribution and may authorize others to do so, for official U.S. Government purposes only, if the U.S. Government contract or grant so requires. (U.S. Government, U.K. Government, and other government employees: see notes at end)

F. COPYRIGHT NOTICE

The Contributor and the company/employer agree that any and all copies of the final published version of the Contribution or any part thereof distributed or posted by them in print or electronic format as permitted herein will include the notice of copyright as stipulated in the Journal and a full citation to the Journal as published by Wiley-Blackwell.

G. CONTRIBUTOR'S REPRESENTATIONS

The Contributor represents that the Contribution is the Contributor's original work, all individuals identified as Contributors actually contributed to the Contribution, and all individuals who contributed are included. If the Contribution was prepared jointly, the Contributor agrees to inform the co-Contributors of the terms of this Agreement and to obtain their signature to this Agreement or their written permission to sign on their behalf. The Contribution is submitted only to this Journal and has not been published before. (If excerpts from copyrighted works owned by third parties are included, the Contributor will obtain written permission from the copyright owners for all uses as set forth in Wiley-Blackwell's permissions form or in the Journal's Instructions for Contributors, and show credit to the sources in the Contribution.) The Contributor also warrants that the Contribution contains no libelous or unlawful statements, does not infringe upon the rights (including without limitation the copyright, patent or trademark rights) or the privacy of others, or contain material or instructions that might cause harm or injury.

CHECK ONE BOX:

☐ Contributor-owned work

ATTACH ADDITIONAL SIGNATURE PAGES AS NECESSARY

Contributor's signature ______________________ Date __________

Type or print name and title ______________________

Co-contributor's signature ______________________ Date __________

Type or print name and title ______________________

☐ Company/Institution-owned work (made-for-hire in the course of employment)

Company or Institution (Employer-for-Hire) ______________________ Date __________

Authorized signature of Employer ______________________ Date __________

☐ U.S. Government work

Note to U.S. Government Employees
A contribution prepared by a U.S. federal government employee as part of the employee's official duties, or which is an official U.S. Government publication, is called a "U.S. Government work," and is in the public domain in the United States. In such case, the employee may cross out Paragraph A.1 but must sign (in the Contributor's signature line) and return this Agreement. If the Contribution was not prepared as part of the employee's duties or is not an official U.S. Government publication, it is not a U.S. Government work.

☐ U.K. Government work (Crown Copyright)

Note to U.K. Government Employees
The rights in a Contribution prepared by an employee of a U.K. government department, agency or other Crown body as part of his/her official duties, or which is an official government publication, belong to the Crown. U.K. government authors should submit a signed declaration form together with this Agreement. The form can be obtained via http://www.opsi.gov.uk/advice/crown-copyright/copyright-guidance/publication-of-articles-written-by-ministers-and-civil-servants.htm

☐ Other Government work

Note to Non-U.S., Non-U.K. Government Employees
If your status as a government employee legally prevents you from signing this Agreement, please contact the editorial office.

☐ NIH Grantees

Note to NIH Grantees
Pursuant to NIH mandate, Wiley-Blackwell will post the accepted version of Contributions authored by NIH grant-holders to PubMed Central upon acceptance. This accepted version will be made publicly available 12 months after publication. For further information, see www.wiley.com/go/nihmandate.

111 RIVER STREET, HOBOKEN, NJ 07030

COMPUTER APPLICATIONS IN ENGINEERING EDUCATION (CAE)

To:	Laura Espinet
Company:	
Phone:	201-748-8884
Fax:	201-748-6182
From:	
Date:	
Pages including this cover page:	

Message:

Re:

John Wiley & Sons, Inc.

Publishers Since 1807

REPRINT BILLING DEPARTMENT • 111 RIVER STREET, HOBOKEN, NJ 07030
PHONE: (201) 748-8659; FAX: (201) 748-6052
E-MAIL: reprints @ wiley.com

PREPUBLICATION REPRINT ORDER FORM

Please complete this form even if you are not ordering reprints. This form **MUST** be returned with your corrected proofs and original manuscript. Your reprints will be shipped approximately 4 weeks after publication. Reprints ordered after printing are substantially more expensive.

JOURNAL: ***COMPUTER APPLICATIONS IN ENGINEERING EDUCATION*** VOLUME_____ ISSUE_____

TITLE OF MANUSCRIPT____

MS. NO.___________ NO. OF PAGES______ AUTHOR(S)______________________________

REPRINTS 8 1/4 X 11

No. of Pages	100 Reprints	200 Reprints	300 Reprints	400 Reprints	500 Reprints
	$	$	$	$	$
1-4	336	501	694	890	1,052
5-8	469	703	987	1,251	1,477
9-12	594	923	1,234	1,565	1,850
13-16	714	1,156	1,527	1,901	2,273
17-20	794	1,340	1,775	2,212	2,648
21-24	911	1,529	2,031	2,536	3,037
25-28	1,004	1,707	2,267	2,828	3,388
29-32	1,108	1,894	2,515	3,135	3,755
33-36	1,219	2,092	2,773	3,456	4,143
37-40	1,329	2,290	3,033	3,776	4,528

** REPRINTS ARE ONLY AVAILABLE IN LOTS OF 100. IF YOU WISH TO ORDER MORE THAN 500 REPRINTS, PLEASE CONTACT OUR REPRINTS DEPARTMENT AT (201)748-8659 FOR A PRICE QUOTE.

COVERS

100 Covers - $90 • 200 Covers - $145 • 300 Covers - $200
400 Covers - $255 • 500 Covers - $325 • Additional 100s - $65

q Please send me __________ reprints of the above article at....................... $____________
q Please send me __________ Generic covers of the above journal at....................... $____________
Please add appropriate State and Local Tax {Tax Exempt No.__________________} $____________
Please add 5% Postage and Handling.. $____________
TOTAL AMOUNT OF ORDER** .... $____________
International orders must be paid in U.S. currency and drawn on a U.S. bank

Please check one: q Check enclosed q Bill me q Credit Card
If credit card order, charge to: q American Express q Visa q MasterCard q Discover
Credit Card No.______________________________ Signature______________________________ Exp. Date___________

Bill To:
Name____
Address/Institution________________________________
____
____
____
Purchase Order No. ______________________________

Ship To:
Name ____
Address/Institution________________________________
____
____
____
Phone______________ Fax _________________
E-mail: ____________________________________

Softproofing for advanced Adobe Acrobat Users - NOTES tool

NOTE: ACROBAT READER FROM THE INTERNET DOES NOT CONTAIN THE NOTES TOOL USED IN THIS PROCEDURE.

Acrobat annotation tools can be very useful for indicating changes to the PDF proof of your article. By using Acrobat annotation tools, a full digital pathway can be maintained for your page proofs.

The NOTES annotation tool can be used with either Adobe Acrobat 4.0, 5.0 or 6.0. Other annotation tools are also available in Acrobat 4.0, but this instruction sheet will concentrate on how to use the NOTES tool. Acrobat Reader, the free Internet download software from Adobe, DOES NOT contain the NOTES tool. In order to softproof using the NOTES tool you must have the full software suite Adobe Acrobat 4.0, 5.0 or 6.0 installed on your computer.

Steps for Softproofing using Adobe Acrobat NOTES tool:

1. Open the PDF page proof of your article using either Adobe Acrobat 4.0, 5.0 or 6.0. Proof your article on-screen or print a copy for markup of changes.

2. Go to File/Preferences/Annotations (in Acrobat 4.0) or Document/Add a Comment (in Acrobat 6.0 and enter your name into the “default user” or “author” field. Also, set the font size at 9 or 10 point.

3. When you have decided on the corrections to your article, select the NOTES tool from the Acrobat toolbox and click in the margin next to the text to be changed.

4. Enter your corrections into the NOTES text box window. Be sure to clearly indicate where the correction is to be placed and what text it will effect. If necessary to avoid confusion, you can use your TEXT SELECTION tool to copy the text to be corrected and paste it into the NOTES text box window. At this point, you can type the corrections directly into the NOTES text box window. **DO NOT correct the text by typing directly on the PDF page.**

5. Go through your entire article using the NOTES tool as described in Step 4.

6. When you have completed the corrections to your article, go to File/Export/Annotations (in Acrobat 4.0) or Document/Add a Comment (in Acrobat 6.0).

7. **When closing your article PDF be sure NOT to save changes to original file.**

8. To make changes to a NOTES file you have exported, simply re-open the original PDF proof file, go to File/Import/Notes and import the NOTES file you saved. Make changes and re-export NOTES file keeping the same file name.

9. When complete, attach your NOTES file to a reply e-mail message. Be sure to include your name, the date, and the title of the journal your article will be printed in.