The GCAR-EAD: Learning Environment with Remote Experiments support

F. M. Schaf, C. E. Pereira
Universidade do Rio Grande do Sul / Electrical Engineering Dep., Porto Alegre, Brazil

Abstract — The paper presents a proposal to integrate mixed reality remote experiments into virtual learning environments (VLEs) using the concept of Interchangeable Components, which can represent either real or virtual devices or software in industrial automation systems. Combinations of real and virtual technical plants and automation systems are used in different learning scenarios for teaching control and automation concepts. Configurations of Interchangeable Components can be dynamically created via a virtual learning environment by configuring database parameters. The proposed system includes a remote web interface that follows a thin client strategy and is designed to be compatible with web browsers including basic Java support. As the architecture that supports the integration of virtual and real components is located on the server side, remote students/users are only concerned with the experiment and do not need to be aware of the system that provides this integration and flexibility. In the current version, interchangeable components are integrated via an OPC – OLE for Process Control – interface, a widely adopted standard in the control and automation area. The proposed approach also provides practical and theoretical support for experiments within a collaborative virtual environment. This paper includes a description of four experiments developed using the proposed environment and concepts.

Index Terms— Remote Experiments, Virtual Learning Environments, Collaborative Learning Environments.

I. INTRODUCTION

The growth of the Internet has brought new paradigms and possibilities in technological education. In particular, it allows the remote use of experimental facilities that can be used to illustrate concepts handled in classroom and serves as an enabling and powerful technology for distance teaching. Through its world wide connectivity, the Internet also allows to have learning materials available to a much larger audience of students, giving them a greater flexibility in terms of defining by their own the speed and sequence of subjects during learning. Local and remote experimentation allows the application and testbeds of theoretical knowledge in practical situations [1]. The use of laboratories supports students activities both in terms of active learning [2, 3], distributed learning [1] and team learning [4]. Web accessible laboratories with remote experiments have become an attractive economical solution for the increasing number of students [5]. They represent a “second best of being there” (SBBT) [6] solution for students and laboratories with expensive equipments. Remote experiments increase the accessibility to laboratory equipment and also provide space and time flexibility, i.e. students can be anywhere anytime performing their experiments via Internet [5]. Following this trend, many institutions around the world have been engaged in the development of Web based experimental settings. Systems aiming at teaching and research in several different areas have been proposed, such as digital process control [19], [18], aerospace applications [19], PID control [20], predictive control, embedded communication systems [17], and real-time video and voice applications [21]. Mostly, these experiments utilize customized devices and software to make small-scale textbook-like experiments remotely available.

Considering education on control and automation systems, a key issue is the reduction of the gap between classical theoretical courses and real industrial practice. Hence, it is important to allow students to operate with devices, systems, and techniques as close as possible to those they will be confronted in industrial settings. Unfortunately, to reproduce in an academic environment a real industrial plant is not an easy task. Industrial equipments are in general very expensive (both in terms of acquisition and also in maintenance costs) and usually require a large area for installation. Furthermore, safety constraints should also be taken into account.

All above-mentioned factors restrict the use of real industrial devices in academic laboratories, which in general are then structured as small-scale experiments with little connection to industrial reality. Within this context, making an industrial lab facility available via Web and therefore accessible - at flexible times - to a larger number of individuals, helps to improve the overall cost-effectiveness of such solution.

However, experience has shown that allow the availability of remote experiments is not a sufficient condition to ensure success in the learning process. Remote lab experiments that are not offered together with learning material explaining the topics that are to be learned in the experiment usually lead students to the use of a “trial and error” strategy with a lower learning impact than expected. Additionally, the fact that remote labs are made available 24/7 for a large audience of students increases the demand in the number of faculty members and tutors that are necessary to provide on-line guidance to students.

In order to alleviate these problems remote experiments can be integrated into virtual learning environments (VLEs) [7, 8, 9] that manage and provide learning materials before, during and after the experimentation. This work proposes such an integrated learning environment, on which mixed reality lab experiments and
student guidance tools are combined for control and automation education.

Mixed reality experiments [10], on which simulated components can be combined to real equipment to provide more practical situations, are used to illustrate different learning situations according to the knowledge level of remote students.

The proposed environment has been developed with the context of the RExNet Consortium [11]. This Alfa project, funded by the European Community, had mainly three goals: to share (1), harmonize (2), and spread (3) current skills on remote experimentation. The first goal directly addresses the essence of the ALFA program (financial support), namely it calls for the cooperation among the consortium partners: those having already available a remote experiment (or lab) grant access to the all consortium, and those not having it should endure all efforts to set up at least one remote experiment useful to the consortium. Harmonization is a direct consequence of having universities from distinct countries with different languages and cultures. Among other items it includes interface harmonization, with support to different languages, and curricula harmonization, i.e., defining a common set of practical experiments for a given course already served by a remote lab (or set of remote experiments). Each university participating in the RExNet project must act as a disseminating party within its own country, i.e., spread the access to remote experiments to other surrounding universities.

The paper is divided as follows: Section 2 describes GCAR-EAD, a virtual learning environment for control and automation engineering education, which integrates mixed reality and remote labs with adequate learning materials. In addition to this, the proposed system includes some student guidance tools, which analyze the results of experiments performed by students in order to identify and suggest topics to be reviewed. The main GCAR-EAD modules are depicted and a detailed explanation on each module and its implementation are included. One of the main underlying concepts of the proposed environment is the so-called “interchangeable components”, modular entities with well defined interfaces which allow a transparent integration between real and virtual modules in the environment. Section 3 describes some experiment scenarios, which exploit different combinations of real and simulated automation system modules (both technical plants as well as industrial automation systems can be simulated), which have demonstrated to be very effective in teaching automation and control concepts to engineering students. Section 4 discusses some results achieved when using GCAR-EAD in undergrad and graduate courses at UFRGS in Brazil. Finally, in Section 5 conclusions are drawn and future work directions are signaled.

II. GCAR-EAD

Our experience in applying remote experiments for control and automation started in 2002 with the construction of a remote laboratory with a Foundation Fieldbus pilot plant [16] (see Fig. 1). With that system, students were able to learn PID tuning control techniques and industrial communication protocols by working with hypertext-based learning material and by remotely accessing the pilot plant in order to perform experiments on which they could put in practice the theoretical concepts learned in classroom. In order to organize the remote access to the remote plant, a tool has been developed, which was responsible for (i) validating users’ access, (ii) scheduling appointments for students to run the experiments, (iii) controlling experiment scheduling, (iv) tracking data related to students’ activities as well as (v) bring the experiment to a well defined initial state before experiments start.

Experiences in using the Foundation Fieldbus pilot plant showed that due to the fact that the learning material was “loosely coupled” with the remote experiment, students were not able to identify which topics to review in case they could not get the proposed experiments adequately done.

In order to overcome those drawbacks, a system called GCAR-EAD was proposed, which supports remote experimentation and mixed reality. The GCAR-EAD has a more complex architecture (see Fig. 2) that additionally integrates learning material manager (also called virtual learning environments – VLEs), educational materials, remote experiments with mixed reality [10, 22, 23], interchangeable components strategy [7], experiment analysis [12] and simple student guidance tools [12]. The proposed architecture has five main modules: learning (didactic) material manager, student guidance system (or student guide), experiment booking, experiment analysis (or experiment analyzer) and experiment manager/interface. Each of these modules is responsible for controlling a specific functionality of the GCAR-EAD. The interaction with each module is transparent, so that students only interact with the VLE.

A central database is the main communication channel among modules. Stored parameters and variables allow inter-modules interactions. Each of the modules has database communication interface for reading and writing of data to the database (see Fig. 3), where a consistency
checking is performed for safety reasons (for instance, to avoid that students may damage laboratory equipment by erroneously configuring them.

Fig. 2. GCAR-EAD high level architecture.

In the sequence, each of the modules is further described.

✓ **Learning Material Manager:**

This module contains all didactic materials of the GCAR-EAD and monitors all students’ interactions. All users are identified via username and password and depending on users’ category and, in case of students, knowledge level, distinct operations are allowed when accessing available learning material. Distinct learning modes are supported: active learning [2, 3], distributed learning [1] and team learning [4]. Active learning skills are justified since, via environment interactions, students can “self-learn” (or self-teach). Distributed learning skill is obviously linked to the spatial flexibility characteristic offered by VLEs Web-accessibility. The most important skill is however related to collaborative interaction, i.e., students teams (or users in general) may work together increasing the knowledge transfer in a common environment. Previously (in normal “presential” courses) it was assumed that teachers were the source of knowledge and they centralized all courses information. The collaborative learning skill is mostly associated with the social constructionist pedagogic line.

The learning material manager plays the most meaningful role in the GCAR-EAD architecture since this system was created to supply an educational (learning) environment attached to remote experiments for the students.

✓ **Experiment Booking Module:**

This module is responsible for controlling the access to experiments by students. Since real experiments are not replicable, booking systems are necessary to organize the use of the real equipments (or entire real experiments) by the students. User/password information stored in the VLE is checked so that only signed VLE users can book/access experiments. Validated users can select one of the available time slots (30 minutes each) for running their experiments.

✓ **Experiment Interface/Manager Module:**

The experiment interface/manager provides a link among the remote experiments and the VLE and must ensure that the right remote experiment interface is available according to VLE set-up parameters. That means, the experiment manager receives from the VLE a reference to the experiment to be executed and “constructs” the experiment providing also a Java Applet interface for data visualization.

Fig. 3. Access strategy and architecture modules interaction.
This module is also responsible to implement the interchangeable components strategy by linking and combining real and virtual components in a learning scenario.

- **Experiment Analysis:**
  The experiment analysis module comprises tools to evaluate the results of a conducted experiment and determine — based on some metrics derived from the experiment results. The experiment data is supplied by the experiment manager and by the VLE in form of reports or is directly stored in the central database.

- **Student Guidance Module:**
  The last module of the GCAR-EAD architecture is responsible for providing student guidance, which means it receives as input the metrics generated by the experiment analysis module and has to determine whether students have achieved the goals defined by tutors/teachers. If not, this module has to indicate learning materials to be reviewed by the students. Therefore, this module acts as a simple tutoring system integrated in the VLE. The guidance must take into account all information stored in the database related to the student (student level, previous experiments performed and visited learning materials). The tutor can also analyze reports (history data) of completed quizzes and VLE proposed activities like assignments, etc.

A. **VLE integration with Mixed Reality supporting Interchangeable Components**

While the remote access of real laboratory equipment has several advantages, there are also some issues to be considered for teaching control and automation concepts:

(i) the number of students / students groups working simultaneously is equal to the number of physical experiments available. Only one student (or group of students) can simultaneously access — generally only one experiment is available to a large group of students, thus systems to organize queues to access the experiment in sequence are necessary.

(ii) long waiting times caused by slow dynamic systems.

(iii) interlocking systems have to be carefully developed in order to avoid that students may damage components via improper actuation.

Two alternatives were identified in order to overcome these drawbacks: (i) use of pre-recorded experiments (ii) use of simulated components.

The use of pre-recorded experiments can be justified due to the fact that it is quite common to have a large group of students having to perform the same assignment within a given time interval. In this case, it becomes quite often that students access to experiment is delayed, even when students would like to execute the experiment with the same initial and working conditions.

Another alternative is the use of simulated components. Simulations, although sometimes unrealistic, have several advantages that can be explored in different learning scenarios. One of the advantages of using simulations is that they can be easily replicated. Students can then simultaneously use multiple copies (replicas) of the same simulation simultaneously, i.e., identical copies of a simulation model can be executed at the same time by various students. The simulation replicas instead of real experiments do not imply on more equipment, because replicas only consume more computer memory and processor power. Another advantage of using simulation is that students can speed up slow dynamics systems for quick visualization using simulation models (for instance, while the real process of heating a tank can take hours, the analysis of aspects such as rising time, overshoot, can be done in seconds using simulations).

By analyzing the pros and cons of real vs. simulated experiments, one can see that in some sense they are complementary so that a combination of both possibilities seems interesting. The so called interchangeable components strategy has been developed to allow this combination of both real and virtual components (see Fig. 4) [7]. The use of interchangeable components enables the definition of a variety of learning scenarios. For instance, real automation equipment such as PLC controllers could be remotely programmed by students and these controllers would then control a simulated industrial plant. This would then avoid the possibility that students with little experience in industrial programming would damage a real industrial plant. Additionally, simulated plants can be used to evaluate robustness of control algorithms when the (simulated) technical plant presents unexpected behaviour. On the other hand, simulated automation systems can be useful to show step by step execution of industrial controllers.

![Fig. 4. Interchangeable components strategy](image)

The use of interchangeable components enables the definition of a variety of learning scenarios. For instance, real automation equipment such as PLC controllers could be remotely programmed by students and these controllers would then control a simulated industrial plant. This would then avoid the possibility that students with little experience in industrial programming would damage a real industrial plant. Additionally, simulated plants can be used to evaluate robustness of control algorithms when the (simulated) technical plant presents unexpected behaviour. On the other hand, simulated automation systems can be useful to show step by step execution of industrial controllers.

---

perturbations, for didactic issues, the implementation of simple models is more adequate in early stages of experiments learning process. In simplified and ideal models the direct application of the theory concepts is an important issue of this first learning scenario.

Step by step execution can be also implemented since simulated equipments are used and real world constraints are easily manipulated. Since the experiment is purely simulated (virtual) some advantages as models replication can be implemented. Thus, multiple fully simulated learning scenarios can be accessed simultaneously and all experiment data can be easily replicated. Security and accessibility issues like booking systems do not need to be addressed for this scenario.

2) Mixed simulated/real components scenarios: This configuration can be used, for instance, in the interaction between a simulated controller and a real plant to elucidate how acquired data from the real plant varies from the ideal model and this can cause instability in the controller programmed logic, consequently, some precautions must be addressed in the simulated equipment to treat that instability. When dealing with a real controller, some problems also occur in the delay of the control logic, since the controller cannot process the acquired data instantaneously (commonly, the controller cycle time is responsible for this delay).

3) Scenario with real components: This experiment scenario is the typical implementation of remote laboratories where SBBT [6] is implemented and students can perform experimentation using real components and observe how theory applies into practical applications. Here, non-linear behavior, perturbations, physical constraints, communication delays, etc, affect the experiment and all these “real life limitations” can be visualized. Obviously, this kind of experiment is not so easy replicated and some access control must be addressed, like booking systems, safety concerns, etc.

The experiment scenario is described by simple parameters stored in the central database. The VLE must only write certain parameters in the database to configure the learning scenario. The envisioned automated interchangeable components selection the following sequence (see Fig. 5): 1. student interacts with the VLE and “selects” the available remote experiment; 2. VLE writes scenario parameters in the central database; 3. experiment manager reads scenarios parameters from the central database; 4. according to the parameters the experiment manager sets “connections” or “links” with the interchangeable components; 5. connection with different real/virtual equipments is established; 6. Web interface to the learning scenario is displayed and the user has interface granted access nested (integrated) in the VLE.

B. VLE integration with Tutoring Systems

The proposed VLE integration with tutoring systems is responsible for every GCAR-EAD interaction feedback. Tutoring systems are dependent to several other tools or modules. Each one of the GCAR-EAD architecture modules stores informations in the central database that can contribute to the tutoring system feedback compilation.

Basically an integrated tutoring system gives two kinds of feedback: (i) allows remote experiment configuration according to the user (student) level, i.e., students with no previously recorded interaction with the experiment should start with basic experiments (usually the fully simulated scenario) while more advanced students can directly go to more complex experiments; (ii) infer didactic material according to student performed experiment.

The first type of feedback compilation, searches in the central database only for previously performed experiments and visited learning materials. Based on this data, it “decides” which type of learning scenario the student has granted access.

The second feedback type uses besides visited learning materials data also metrics or reports generated by the experiment analyzer to suggest specific didactic material to the student.

The experiment analyzer plays a meaningful role in the experiment-driven tutoring system feedback. There are two proposed types of experiment analyses: (i) for dynamic experiments the result of the analysis (“evaluation”) is mostly computed off-line, that means after the experiment has been concluded control metrics like overshoot and rise time are calculated; (ii) on the other hand, discrete experiment based on logic control can be evaluated in execution time, since the digital I/Os can be tested while the experiment is running. The first type is called pos-runtime- while the other runtime-analysis, but both produces reports that are stored in the central database.

The post-runtime-analysis tool compares the results of experiments performed by students with the requirements.
specified in the experiment assignment. Usual control systems metrics such as maximal overshoot, rise time and settling time are adopted as comparison criteria. These metrics are usually applied in control systems theory and are directly related to the performance of remote experiment’s behaviour.

The process flow of a post-experiment analysis is as follows (see Fig. 6): 1. student interacts with VLE and “selects” the available remote experiment; 2. VLE writes scenario parameters in the central database; 3. experiment manager reads scenarios parameters from the central database; 4. experiment is performed by the student; 5. experiment manager generates a report with important and relevant experiment data; 6. post-experiment analyzer reads report; 7. metrics evaluating the experiment are generated; 8. tutoring system analyses (compares) metrics against requirements defined in problem specification; 9. based on the metrics tutoring system suggests learning material if experiment’s goals were not reached.

The runtime-analysis tool somewhat simpler, it only analyses certain logic rules in the experiment and if these rules are not met “warning messages” are shown with the problems that occurred. This tool is envisioned to be “built-in” in the experiment interface and works in a similar way as the post-runtime-analyzer. Although the evaluation is made in the execution time, the suggestion is also post-experiment. Stored experiment messages (experiment report) are read by the tutoring system and learning material suggestion is done in case experiment’s goals were not reached.

III. GCAR-EAD IMPLEMENTATION

This section will describe some applications developed using the GCAR-EAD architecture presented in this paper. Most of the applications were developed within the scope of the RExNet Project [11].

The learning materials manager module offers a user-friendly interface with the free, open-source course management system called MOODLE [29]. MOODLE is a widely used software package designed with social constructionist pedagogical principles to help educators create effective online learning communities with many collaborative learning tools. MOODLE was chosen as VLE in the RExNet consortium, and all courses and learning materials are organized to provide maximum intuitive perceptions, simplicity and knowledge transfer.

OPC-DA [14] standardized interface is employed in order to allow scenarios configuration without building four different and distinct experiments (three learning scenarios). The OPC interface provides a common, simple and reusable interface to the interchangeable components strategy implementation. Basically the same interface is used for students interaction with real and simulated components (and therefore they are considered “interchangeable”). Each of the components (real/simulated plant and controllers) must have an OPC interface so that automatic selection of the interchangeable components can occur, consequently, scenarios configuration can be automatized. Given that OPC is widely used in automation area, most of
simulators and data acquisition systems used in industrial automation systems generally include support for OPC-DA server applications. Therefore, OPC works as “glue” among different applications (in this case, components).

The experiment booking, student guidance and experiment analysis modules are implemented in PHP code to simplify the integration and the interaction with the MOODLE software. A simple SCADA software, named Elipse SCADA [13], is used as experiment manager and is responsible for providing connections with the real/simulated equipment and to supply Java Applets as experiment visualization interfaces. Elipse SCADA also has interfaces to deal with ODBC (with MySQL connector) and with OPC servers.

All developed remote experiments (case studies implementations) follow common software architecture (see Fig. 7) with the above mentioned softwares, the Apache web server software, MySQL as database interface manager, OPC-DA for experiment level communication interface and the Isagraf [15] as simulation tool for all cases.

The GCAR-EAD uses the stored login/password of MOODLE to grant access to the remote experiments. If the user (student) is not signed in the MOODLE, his access to the remote experiments is refused. A booking system user-friendly interface was also developed in order to organize students access in different time slots.

IV. DEVELOPED REMOTE EXPERIMENTS

According to our developed remote experiments, four major courses for control and automation education were developed in the GCAR-EAD environment organized by the MOODLE VLE (see Fig. 8): (i) PID Controllers, (ii) IEC 61131 Standard for programming industrial automation applications, (iii) Foundation Fieldbus Communication Protocol and (iv) Remote Experiments Course. Other related courses are in development, e.g. a complete course of control systems theory (undergraduate course in electrical engineering) in the UFRGS.

A. Enhanced Foundation Fieldbus Pilot Plant

The first experiment was developed using a Foundation Fieldbus Pilot Plant [16, 7] (see Fig. 1), which has been used as a remote laboratory for several years at UFRGS for teaching PID controller theory, in particular how to tune PID parameters. Using MOODLE, the original remote laboratory was integrated into the GCAR-EAD by including new learning material on PID control theory as well as incorporating an experiment analysis tool to calculate control metrics from the results of the experiments performed by students. In this experiment students have to control the water level in two tanks by acting on pumps and valves (see Fig. 9). Additional to a course on “PID Controllers”, some other courses explaining how to use the experiment as well as a course of the Foundation Fieldbus industrial communication protocol were included.

The interchangeable components strategy is implemented in Elipse basic scripts in the SCADA software. MOODLE PHP programs write scenario parameters and scripts configure the experiment setting up links and connections between selected components.

B. Thermal Plant

The second experiment uses a simple thermal plant built with a PID industrial controller and simple electronic equipment to illustrate temperature control techniques and the use of industrial controllers. The experiment consists on a thermal resistance and a PTC100 thermal sensor connected to an industrial controller N1100 from Novus [25]. This controller manipulates the electric current that flows in the resistance via PWM switching and thus affecting on the temperature of the resistance. This industrial controller offers PID control technique. Students must, based on open loop trails, identify and control the temperature (in closed loop) of the experiment setting PID parameters given temperature setpoints.

Again, special courses were elaborated and similar student guidance and experiment analysis tools (see Fig. 10) were developed. This experiment was easily adapted in the environment due to the reuse of modules previously developed for the Foundation Fieldbus plant.
C. Mixed Reality Electro-Pneumatics Workbench

The third experiment, a mixed reality workbench for teaching mechatronics (electro pneumatic) for industrial apprentices, was developed in collaboration with researchers from the Universities of Bremen and Berlin [10], members of the RExNet consortium. In this case, the so-called deriveSERVER developed in Germany, is integrated via an OPC interface to the GCAR-EAD environment [9]. This system is very flexible and has great interactivity with the apprentices (see Fig. 11).

The system provides mixed reality experiments through the use of “hyperbonds”, which allows a tight coupling between physical and virtual phenomena [10]. Physical signal such as air pressure, electric potential, etc. are converted into binary logic information and vice-versa. Hyperbonds are based on the Bond graphs theory [26] which provides a unified view on different systems using the notion of effort and flow. The bond graph theory has been, then, further developed by Karnopp et al. [27].

The system software architecture is entirely based on client-server modules allowing great modularity and mobility (system distribution). ROMAN (Real Object Manager) is the software that works as master (server), responsible to bridging (interconnect) all system’s interactions. The clients (modules), called ROMAN plug-ins, are: (i) VCK (Virtual Construction Kit) and (ii) hyperbond (software and hardware). The plug-ins communicate with the ROMAN via communication sockets using a specific ROMAN-protocol.

The hyper-bond software interface was modified from the original German project. It supports now, besides FESTO [28] EasyPort hardware connectivity also OPC and parallel/serial communication interfaces. Thanks to the client-server architecture (communication sockets) of the hyperbond to the experiment manager (ROMAN), students can use local hardware attached to simple parallel PC port to interact with the experiment and also local OPC servers with any kind compatible equipment (see Fig. 12).

The new hyperbond interfaces offers remote hardware or simulator interaction using any of the interfaces, i.e., students can have their own hardware/software (compliant with the new interfaces) interacting remotely with the system. SENAI’s PLC logic simulator (simple ladder simulator), named Relés, was modified to address his inputs and outputs directly to the PC parallel port. With this modification, Relés simulator can “transport” his I/Os to the Web-accessible virtual workbench through the hyperbond interface. The same communication “bridge” can be accomplished with an OPC-DA compliant software or hardware.
In order to focus the system utilization by students task oriented experiments scenarios were developed so that students have a well specified goal.

D. Simulated Bottle Production Plant

The fourth experiment is a simulated bottle production plant [12], whose behaviour is modeled in the ISAGRAF software using IEC61131 programming languages. This experiment provides a very didactic and reusable experiment that can be combined to the other experiments presented, forming a complete combined experiment. For instance, interactions with the mechatronics workbench produce a flexible way to control the experiment and also to integrate with other external OPC servers (other simulations). This experiment has a built-in analysis tool, integrated into the simulation model, which can check the behaviour of automation systems developed by students that control the bottle production process.

All these experiments can be combined using the concepts of interchangeable components, allowing the definition of a very interesting mix of experiments, on which different concepts can be explained and demonstrated.

V. EXPERIENCES USING GCAR-EAD

All case studies have been successfully applied into undergrad and graduated courses on “Control System Design”, “Industrial Automation”, “Time Discrete Control”, etc. The obtained results have been very positive. In particular, one can see that student’s motivation is increased when using remote labs embedded into VLEs. Analysis of logging data shows that while some students access the remote experiments late at night, others prefer to work early in the morning, that means, each one can define their preferable working time and therefore the system is being continuously “tested” and improved with lots of students/teachers suggestions.

<table>
<thead>
<tr>
<th>What was your impression of the course offered in the GCAR-EAD?</th>
<th>Excellent</th>
<th>Good</th>
<th>Regular</th>
<th>Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>43%</td>
<td>43%</td>
<td>0%</td>
<td>14%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>What was your impression of Remote Experiments offered within the GCAR-EAD?</th>
<th>Excellent</th>
<th>Good</th>
<th>Regular</th>
<th>Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50%</td>
<td>43%</td>
<td>0%</td>
<td>7%</td>
</tr>
</tbody>
</table>

| In your opinion which is best for teaching: simulated or real experiments? | Simulation | Real experiments | Both | Combination of both |
|---|---|---|---|
| | 0% | 14% | 36% | 50% |

| Which of the following characteristic(s) are more important in the GCAR-EAD? | Time flexibility | Spatial flexibility | Integrated learning material | Collaborative environment | Internet search integration |
|---|---|---|---|---|
| | 65% | 50% | 14% | 57% | 43% |

Currently a class of 53 students is using the environment for enhancing normal course of control theory (undergraduate course in electrical engineering) lessons. The system is having excellent results since the interactivity of the students is being recorded and can be evaluated. A custom quiz was developed to “evaluate” system qualities and faults according this class of students. The most meaningful questions of this quiz and the answers are shown in Table I.

VI. CONCLUSIONS AND FUTURE WORK

This paper has presented the GCAR-EAD environment, which has the following characteristics:

- allows an integration of mixed reality experiments with virtual learning environments;
- introduces the concept of interchangeable components, allowing multiple combinations of virtual (simulated) and real technical plants and automation systems, which can be used in different learning scenarios;
- includes experiment analysis tools, evaluating the results of experiments performed by the students, trying to infer whether they correctly applied the learned concepts;
- provides student guidance through the learning material, helping students to identify topics to be reviewed in order to fulfill the goals of the assigned experiments.

While the proposed environment has proven to be very useful for control and automation education, there are still some challenges to be faced:

- the synchronization in the timing behaviour of the virtual and real equipment is dependent of the communication delays in the network infrastructure. In the current implementation, this delay is of around two seconds for the whole communication between client and the end actuators, even in Intranet communication. While this is OK for technical plants with slow dynamics (what is the case in the selected experiments) it has to be improved. Of course there is a trade-off in having geographically distributed applications and the higher communication times that are required;
- while in its current version the GCAR EAD environment does allow the configuration of mixed reality experiments, tools with a higher level support to tutors is necessary in order to ease the definition of complex experiments.

The proposed environment can also be used for collaborative engineering since experiments can be distributed into several sites and several students (users) can interact using the same environment.

It is widely believed that collaborative experiences are powerful drivers of cognitive processes and can significantly enhance learning efficiency. The benefits of collaborative learning are widely researched and advocated throughout literature [24]. Regardless of the
varying theoretical emphasis in different approaches on collaborative learning (e.g. social constructivism), research clearly indicates that in many (not all) cases students learn more effectively through collaborative interaction with others. This motivates to prepare remote labs for collaborative learning and to use them in distributed teaching scenarios with simulation tools, hands-on laboratories and practical workshops. There is a strong demand for research that seeks to create such a “blended learning” [30], where collaborative remote labs can play a significant role. Emphasis on collaboration adds new technical requirements to the design of remote laboratories. As a whole, there is a necessity to improve the usability of collaborative remote laboratory tools because otherwise learners may quickly get frustrated and stop working with it.

ACKNOWLEDGMENT

This project has been financially supported by the Brazilian research agencies CAPES, FINEP, and CNPq. Thanks are also given to SENAI in Brazil for the financial support to the development of the mechatronics experiment. Authors also would like to thanks to their financial support to the development of GCAR-EAD.

REFERENCES


AUTHORS

F. M. Schaf is with the Universidade do Rio Grande do Sul, Oswaldo Aranha 103 - Porto Alegre, Brazil (e-mail: frdms@ece.ufrgs.br).

C. E. Pereira, is with the Universidade do Rio Grande do Sul, Oswaldo Aranha 103 - Porto Alegre, Brazil (e-mail: cpereira@ece.ufrgs.br).

This work was supported in part by Brazilian research agencies CAPES, FINEP, and CNPq.