Abstract - We introduce a distributed environment to learn by observing others and the consequences of their behavior. Students learn by competing with other students, comparing experiences, and then adopting a new behavior. The environment integrates a novel communication model, a virtual laboratory and an intelligent tutor. It was applied in a virtual robotics laboratory, where the students learn by doing and watching other people’s work, in a robotics competition. The competition is related with actuators and control theory. The students design a control program, taking care of the mechanical and sensor aspects, which were explored previously in the learning enviornment. To participate in a contest, the students load their program, which is executed in the virtual lab. Up to 4 participants, as well as additional observers, watch the robots performance in the simulated environment. We conducted a user study with 20 students, and compared the learning gain of students that participated in competitions, against other that practiced individually. Preliminary experiments show that this environment motivates and improves student learning.

Index Terms – collaborative learning, virtual laboratories, robotics, and computer supported collaborative work.

INTRODUCTION

Current research on Computer Supported Collaborative Work (CSCW) as tools for learning is based, to a large extent, on three components:

a. A piece of intelligent software built to mimic, portray, or characterize some sort of human intelligence.

b. Artifacts that allow for distributed and virtual communication-interaction to take place.

c. A pedagogic strategy that sustains the presentation of educationally content, and the semantics of the interaction between humans that exhibit imperfect knowledge and skills on a fixed domain.

There are several problems for modeling and developing collaborative learning. It is not one single mechanism: peers do not learn because they are two or more, but because they perform some activities which trigger specific learning mechanisms. But, in addition, the interaction among subjects generates extra activities (explanation, disagreement, mutual regulation, etc.) which trigger extra cognitive mechanisms (knowledge elicitation, internalization, reduced cognitive load, etc.) [1].

Learning is a social activity, so learning environments that consider and promote collaboration (or competition) are one of the most powerful learning tools. We are interested in exploring how students work together and learn from each other, by developing collaborative and competitive learning environments. Additionally, edutainment creates an engaging learning experience that typically seeks to instruct or socialize its audience by embedding lessons in some familiar form of entertainment: television programs, computer and video games, films, music, websites, multimedia software, etc. [2]. Contests give the opportunity to rewarded learning and create something special as a result of an educational experience.

In this work we introduce a shared learning environment to allow people to learn by competing with other, comparing experiences, and then adopting a new behavior. The learning environment includes: (i) a communication model, (ii) a virtual laboratory, and (ii) an intelligent tutor. The model was applied in a virtual robotics laboratory, where the students learn by doing and watching other people’s work, in a robotics competition. We have done two user studies based on this environment, with groups of students in an undergraduate robotics course. Students practiced in a virtual robotics laboratory, and part of them participated in a competition. The results show that student that were engaged in the competitions have higher grades in a post test related to the relevant concepts excerciced in the experiments.

SHARED LEARNING ENVIRONMENT

The shared learning environment compromises 3 main elements:
1. A communication model that allows for interaction among participants and observers in a distributed competition.
2. A virtual (simulated) laboratory where the participants train and compete.
3. An intelligent tutor that guides the students in the training phase towards a competition.

Next we describe each one of these elements.
The Centralized Publishing Model for Interaction (MPCI) is a communication model designed to support the communication between modules that have a high interaction, exchange a large amount of data, need that the calculation takes place on one site and that the addition of a new module does not affect the communication cycle already established [3]. The MPCI model is built to support the distributed contest for learning mobile robotics, and it is based on two communication architectures: client-server and publish-subscribe.

Client-Server Architecture

In this architecture, one or more clients and one or more servers form a system that allows distributed computation, analysis and presentation. In a system like this, a client is a module that interacts directly with the user and translates the user commands on queries that are sent to the server via the communication system. Then, it performs data analysis on the commands or queries sent back from the server to present to the user the data over the user’s interface. Meanwhile, the server provides a service to the client responding to the client queries or commands. In such a system, clients are not allowed to communicate between them, in order to send commands to other clients. A given client must send them via the server and wait for the answer that will come through the server again [4].

Publish-Subscribe Architecture

In this architecture, each system entity exchanges a set of messages in order to participate on a decentralized communicating process. Each component is able to produce a message (or data) and announce (publish) that this data has been produced the other components can consume it. Other components can be waiting (or subscribe to) the production of new data. Messages are not limited; they can be character strings or more complex objects, complex processes, servers, applications, tools or even an instance of a system. The main advantage of this architecture is that any component can achieve a subscription to a data without knowing the identity of the publicist [5]. The MPCI model is base on the list-based variation of this architecture [6].

Communication Cycle

The components interaction deals with the data exchange organization between servers/publishers and clients/subscribers. It is based on publish-subscribe mechanism, shown in figure 1. To start the cycle, the server/publicist must tell to the others that it can receive subscriptions (Step 1). Next, clients/subscribers can request a subscription to the server/publicist using the mechanisms provided by the publishing media (Step 2).

If the server/publicist needs some information from the client to start the calculations, the client must send it and the server will store it in its local database. (Steps 3, and 4). After recollecting this data, the server/publisher can start the calculations (Step 5). When this is finished, the server/publisher must publish and notify the clients/subscribers (Step 6). When the client receives the publish notification, it will use the new data to make some local processing and then present the result to the client (Steps 7 and 8). These last steps can be repeated if the server must continue its processing.

If a client/subscriber needs to include itself in the cycle, it only has to subscribe to the server/publisher. To leave the cycle it only has to rescind the subscription. These two events does not affect the processing because the server/publisher does not need to know the client, all the communication and the data sending is done through the publishing media who deals with this kind of issues.

This model offers several advantages:

- The use of thin clients. To modify the application someone has only to change the server and the client can remain as is.
- If the number of clients grows or decrease, the application is not affected, it is not needed new software because the publishing media controls the subscriptions allowing the server only to focus on the calculations and the clients on the presentation.
- If a client misses the connection, the processing can continue because the server is not aware of the clients that are subscribers of its data.
- There are savings on the storage space because the data only exits in the server; the data are only presented in the client.
- The clients receive the data at the same time, this feature is very important in simulated system like a virtual laboratory.

The main limitation of this model is that it is centralized, so if the server goes down, all the application fails. However, the server can be duplicated to make it more reliable.

VIRTUAL LABORATORY

The communication model has been integrated with a virtual laboratory to provide an environment in which several students can interact and learn by doing. A virtual laboratory provides a way to perform experiments with equipment that is not physically present in the same place as the user [7]. There are two types of virtual labs: (i) remote and (ii) simulated. A simulated lab has a computer model of the equipment and its environment, and the users interact with these models to

1-4244-1084-3/07/$25.00 ©2007 IEEE   
October 10 – 13, 2007, Milwaukee, WI
37th ASEE/IEEE Frontiers in Education Conference
F3G-16
perform experiments. In a virtual laboratory, different users can share the system resources and use them at the same time. Generally, these resources are the same for all the students. In addition, if it is necessary that a group of students collaborate in a same experiment, the MPCI model allows the connection of the clients to the shared data.

The virtual lab is part of generic architecture for virtual laboratories that is described in [8]. The main elements of this architecture are a module for initial categorization, a semi-open learning environment, a simulator and an intelligent tutoring system. Next, we briefly describe the tutor.

**INTELLIGENT TUTOR**

We coupled an intelligent tutoring system to the virtual laboratory. The tutor follows the exploration and performance of the student in the lab, updates its model, gives the appropriate help if required, and defines the next experiments. When a student performs an experiment in the virtual lab, the student model propagates the evidence from the experiments’ evaluation to the knowledge objects in the knowledge base. Based on this evidence and the accumulated evidence from previous experiments, the behavior and performance module updates the student model. After each experiment, the results are used by the tutor module to decide the best pedagogical action.

The main component of this tutor is the PRM-based student model [9]. The PRM allows define the main objects involved in the domain [10]. As shown in figure 2, the main classes related with students and experiments were defined. For each class, a number of attributes (information variables and random variables) is defined. For example, the class X4, Experiment results, includes attributes such as id, number of repetitions, success, efficiency, performance. The dependency model is defined at the class level, allowing it to be used for any object in the class. Once the model is specified at the class level, including the attributes and their dependencies, we can extract a skeleton, that is, a general Bayesian network model for a fragment of the model. A skeleton obtained from the model is depicted in figure 3. This network includes the dependencies between the student knowledge at different levels of granularity, and the results of the experiments in terms of performance and exploration results.

The evidence from the experiments results is propagated first to the basic concepts (knowledge items) related to this experiment, and then to the sub-themes and themes according to a hierarchical knowledge structure for the course. This propagation is done in a Bayesian network derived from the abstract model in figure 3, using standard probability propagation techniques. After each experiment, the results are used by the tutor module to decide the best pedagogical action, such as help or lessons. More details are described in [11].

![Figure 3](image)

**CASE STUDY: MOBILE ROBOTICS VIRTUAL LABORATORY**

The shared learning environment was applied to a mobile robotics virtual laboratory where up to four students can interact in a line following simulated contest. Others (students, teacher) can observe the contest. First we describe the virtual robotics laboratory and the available experiments, and then the contest scenario.

**Virtual Robotics Laboratory**

We consider a line following experiment, which requires some basic knowledge in mechanical design, sensors, control theory and programming from the students. We defined a sequence of specific experiments to allow students to train toward automatic control programming before the contests. The experiments are the following:

1) The first experiment involves mechanical properties of robotics design concepts, as shown in figure 4. The educational goals are: (i) to learn mechanical aspects of mobile robotics, and (ii) to practice with kinematical models using manual control.
2) The second and third experiments are designed to explore basic properties of infrared (IR) sensors (which help to change speed and direction) as shown in figure 5.

3) The fourth experiment is related with actuators and control theory. We defined a set of basic robotics instruction for controlling a mobile robot, and constructed an interpreter to simulate an automatic control program. The student needs to write his/her control program previously, taking care of the mechanical configuration and sensors’ positions defined in experiments 1 to 3. To use the virtual laboratory the students need to load his/her control program and the system verifies the syntax of the program. If there are no errors, they can select the execute button and the simulator shows the robot movements based on the control program.

Initialization

In order to participate in a contest, a user must sign in and get a category from the intelligent tutoring system. Due the fact that this environment is available in a Web environment, there is a possible learning heterogeneity between the users of the system. Moreover, it is necessary to homogenize these users so that they can compete with other students that have similar knowledge. Because of this, it has been decided to classify the students in three categories: Beginner, Intermediate and Expert.

To achieve this categorization, the student academic background can be considered (when used within a formal course), or a quiz that evaluates the knowledge items and concepts that will be used over the virtual laboratory experiments. The academic background or quiz results are used by the initial categorization model (that uses a Bayesian network [12]) to infer the initial categorization for each participant.

The Contest

The contest is based on a simulation of a line following contest in mobile robotics. It consists of 6 stages that are explained next.

i. In order to start a contest, a professor must define it. So the professor must set the date and time of the contest and its category. From this information, the competition process continues as follows. Ten minutes before the starting time, students that have the same category as the contest can subscribe to it. This corresponds to the subscribing process in the MPCI model., as it is shown in figure 6

ii. Students have to define the robot characteristics and write a program to control the robot. The instructions for this program are based on the LEJOS language for mobile robots [13]. After this, the program is compiled by the system and it must have no errors in order to continue.
iii. To simulate the contest, the student must choose the kind of robot (car, tricycle or synchronized), its dimensions (wheel radius, width and length) and the positions of three infrared (IR) sensors, as it is shown in figure 7. Next, this is sent to and stored in the server.

iv. When the contest is about to begin, a circuit is chosen at random between a set of circuits previously defined. Then, the characteristics of the participant robots and the circuit are published so the clients can present them to the students. Now, the contest can start.

v. During the contest, the server creates a clock signal. Each time period, the simulation process will calculate the position of the robots and its sensors’ values. Then, these values are published. In the user interface, the data are displayed according to the student. The student that is participating in the contest will see his/her robot, a sensor panel and the robot dynamic values (speed, state, etc.). The robots of the other participants will be only displayed without the sensor panel and the dynamic values, as it is shown in figure 8.

vi. For the contest to finish, one of two conditions must be present: the time limit is over or the robots have completed the circuit. After this, the intelligent tutoring system decides the winner and displays advice to each participant student.

Anyone with a username and a password in the system can observe a competition. He/She only needs to join the contest and it will automatically receive the contest data. If the observer is a student, only the circuit and robots will be displayed, without the information panels. If the observer is a professor, all the panels will be displayed.

EVALUATION AND RESULTS

We applied an initial evaluation of the contest environment using the centralized publishing model for interaction during the last academic term of 2006, at ITESM Campus Ciudad de México. The contest was evaluated in an undergraduate course with 25 students, with majors in Computer Science or Electrical Engineering. At the beginning each student answered an initial questionnaire to have an initial categorization. The students received training by experimenting in the virtual laboratory. The students were re-categorized during these training sessions. The professor registered six contests. Four students participated in each contest and the other 21 students watched the contest development. Competitors and observers said that they were motivated by the contests to do a better job and they needed to dedicate more time for studying. Observers said that they obtained better solutions after they saw the participants’ solutions during the contests. Students said also that they needed deeper knowledge of robotics to win the contest.

We have concluded a second user study during first term of 2007 to evaluate the impact of the competition on learning. We performed this study with a group of students on a robotics course at the undergraduate level. All the students experimented with the virtual lab individually, but only half of them participated in competitions. Next we describe the study and present the results.

Participants. The subjects were university students between 6th and 9th semester with majors in electrical engineering, computer science or mechatronics. A total of 20 subjects enrolled in a robotics basic course participated in the study. We divided them randomly into an experimental group that used the distributed environment participating in several competitions; and a control group that only practiced with the individual experiments in the virtual laboratory.

Experiment design. All subjects practiced the concepts in mechanical design, kinematics, sensors and control with the individual experiments (experiments 1, 2, 3, and 4) in the robotics virtual laboratory. Both, control and experimental groups, used the lab performing experiments for about 60 minutes. The experimental group had the opportunity to participate in several contest, while the control group only observed the contests.

Post-phase. After the experiments, a post-test was applied to all the subjects. It consisted of 10 questions related with the knowledge objects practiced in the experiments, and of a ten item questionnaire targeted at students’ opinions about their virtual laboratory experience. In addition, the system produced log files that capture the sessions at the level of basic exploration actions and experiment performance results.

Figure 9 summarizes the results for the control and experimental groups of the second study. It shows the grades in the post-test per group, in ascending order per student. We observe an improvement in learning for the students that participated in the contests with respect to those that did not. Additionally, based on a questionnaire, most of the students...
consider that the virtual laboratory is useful to learn the relevant concepts, and 90% said that participating in contests improve their motivation.

![Graph showing grades improvement](image)

**FIGURE 9**
A COMPARISON OF THE EXPERIMENTAL (PARTICIPATED IN SEVERAL CONTESTS) VS. THE CONTROL (WITHOUT CONTEST) GROUPS IN TERMS OF THE RESULTS OF A POST-TEST AFTER EXPERIMENTING IN THE VIRTUAL LAB. THE GRAPH SHOWS THE STUDENTS’ GRADES FOR THE KNOWLEDGE OBJECTS PER STUDENT, IN ASCENDING ORDER.

**CONCLUSIONS AND FUTURE WORK**

Practical experimentation in the laboratory is a key element in engineering education. Contests give the opportunity to be rewarded for learning and create something special as a result of an educational experience. We presented a new learning environment to allow students to interact and observe others and the consequences of their behavior in a contest scenario. The students improve their learning comparing their experiences to the experiences of others, and then adopting a new behavior. This environment has 3 main elements: (i) a communication model that allows for interaction among participants and observers in a distributed competition; (ii) a virtual laboratory where the participants train and compete; and (iii) an intelligent tutor that guides the students in the training phase towards a competition.

The model was applied in a virtual laboratory for mobile robotics, in particular in a line following contest scenario, where the students learn and practice the basic concepts in mobile robotics: mechanical design, kinematics, sensors and control.

We performed two user studies in an undergraduate robotics course; a first one with 25 students, with majors in Computer Science or Electrical Engineering, and a second one with a group of 20 students. In the second study part of the students had only individual experience in the virtual lab, while the other part participated in several competitions. Post-test results show that students that participated in the competition had better grades. However, it is difficult to tell if this is due to what they learn from others in the competitions, just to improved motivation, or to other factors. Further studies are required to confirm these findings, and to have a better understanding of the factors that influence this learning gain.

In the future we will like to explore the interplay between competition and collaboration in virtual laboratory environments. We are designing a collaborative environment in order to allow student teams to design and develop their mobile robots, and then participate in distributed competitions.

**REFERENCES**


