

Presential-like e-learning tool for teaching bioinformatics
Sergio Ramírez⁽¹⁾, Javier Rios⁽¹⁾, Oliverio Vega⁽¹⁾ and Oswaldo Trelles⁽¹⁾
⁽¹⁾Computer Architecture Department, University of Malaga, Spain

MOTIVATION:

E-learning infrastructure is becoming the standard of fact for deploying courses and training material in several domains. Bioinformatics, one of the most dynamic fields of research, is a good example. However, face-to-face interaction between teachers and students is also needed to promote and improve the controlled interchanging of ideas in a geographically distributed environment.

There are many tools that can be used for knowledge distribution, like forums (1) or virtual campus (2, 3), but these tools does no offers the same interaction as needed in a real classrooms. The current video-conference tools offer a more interactive way to talk with a group of persons, but are more oriented to environments where a group of people wants to talk and freely interact with the rest of the group. These video-conference tools do not make distinction between users, all of them considered partners with similar privileges.

In a classroom the converse situation is present: the professor acts as the leader to keep an order; especially if we talk about a very young students. To make this possible it is necessary to assign different roles to users, giving special privileges to teachers to control the participation of students, control the session and assigning temporary rights to the students so can participate interactively in the lecture.

BACKGROUND:

Several tools have been implemented to deal with this problem. For example, GroupLog (2) and Moodle (3) are used to distribute remote knowledge in a virtual campus. Theses tools offer a framework to access the educative material, make examinations, etc but they lack on interaction of a real classroom.

Netmeeting (4, 5), MSN Messenger (6) or Marratech (7) are well-known and profusely used tools in the field of video-conference communications. In addition to the problem described in the previous paragraph, these tools have strong dependency of external servers where the users must be registered and get login. The access of these servers is restricted, so is not possible to perform administrative tasks to configure the working scenario even more, users can communicate with people that are not in the classroom and who can interfere in the development of the lecture.

There are other video conference tools covering the educational environment, (e.g. RELATE: REmote LAnguage Teaching (8)), is an application oriented towards language teaching through the network providing tools for video and audio transmission, blackboard and chat. In the same line, the Pidgin group (9) has developed a new plug-in for their 'pidgin instant messenger' that manages the conversations as if they were a classroom. This tool: Virtual Classroom Plug-in for Pidgin (10) is still under development, but already offers a chat and a blackboard tools to communicate the users and it will have support for video and audio transmission in the near future.

METHODS:

Our main goal is to develop an application that combines the features of the e-learning applications with the interaction that the video-conference tools offer. As a novel contribution, the application grants different rights for different roles to be played by teachers and students so they can have different views of the same environment. The application has a modular design based on the integration of plug-ins, so it can be easily extended with new features.

To have a fluid communication between the teacher and the students is a basic requirement for obtain good teaching results. We have used a basic server-client architecture, that organizes the users in individual classes, totally independent between them.

This schema of connection reduces the interchanging of unnecessary data outside of the classroom and avoids possible interferences between simultaneous classrooms. Inside a classroom the teachers are in charge of distribute the information sent by a student to the rest of them (see FIG. 1). This is possible because the control

of the communications always resides in the teacher who grants privileges to students to participate in the lecture.

Before to begin a lecture, the clients must know some information about the rest of partners, especially who is who and where they are located. The real distributed systems use flooding techniques (11) for solve these problems. These techniques are based on the interactive discovering of new nodes, so its takes time to discover the complete set of clients. This feature make difficult to use it in dynamic systems as this, where the clients can connect and disconnect at any moment. For theses reasons, the system includes a light-weighted server to help people to meet each other, authenticate users, and others basic coordination tasks. It also can be used to recover information about the users: classrooms where they can teach/attend; list of users, etc.

A plug-in based architecture has been applied in both, client and server program to offer an easy mechanism to extend the application with new features. To assist in develop of new plug-ins it is offered an Application Programming Interface (API) for accessing the resources of the application. Using this API, a plug-in can access the graphical interface of the program and request for the creation of new frames where display information, add new options to the menus, etc. To allow the communication between others clients we have developed our own communication-package system. The format of these packages includes a header with the basic information for routing it to the receiver, while the body is left to disposition of the plug-in requirements, so it can implement its own communication protocol.

The application also includes an event system that notify application about the changes in the status of the application like the connection of new users, change on the status of the client (from disconnected to connected ...), etc.

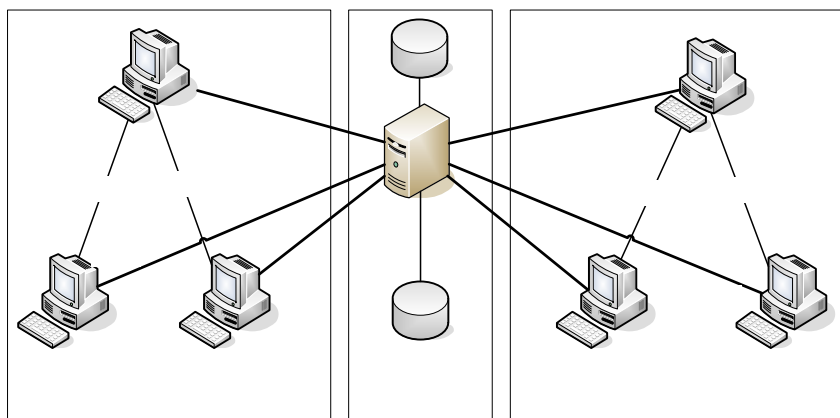


FIG. 1 Client-server layout: It is worthy to observe that clients are only in contact with the main server (for authentication, consistency control, etc) and with their classroom mates through the professor.

RESULTS:

As result, an application has been developed to emulate the interaction with a teacher that offers a real classroom. The application is composed by a central server that is in charge of small task of coordination and a client that focus the rest of features provided. Both of them have been equipped with a plug-in system that facilitates the extension of its features. The client and server provide a base system where the rest of plug-ins can be executed. The system includes functions for sent and received packages of data, maintaining of the graphical interfaces and handling of the client states.

During its execution, a client goes through different states of work. At the beginning the application is not connected to any server, and when the user login in any server, it pass to the connected status. In this status the plug-ins active can sent and receive packages from the central server, as for example the list of classes that the user has, and the list of users and their state that are related with the classroom (Teachers for students, and all for Teacher). When the teacher decides that there are enough students, the lecture can begin (A student must always wait for the teacher to enter in the virtual classroom). Once time the lecture has begin the application pass to the most functional state (InClassroom), in this state the teacher can communicate directly with the rest of the users

in the classroom, and the students through the teacher who is in charge of forwarding the received messages to the rest of the students.

In the FIG. 2 is shown a schema with the status of the application and the events that produce the changes on the status:

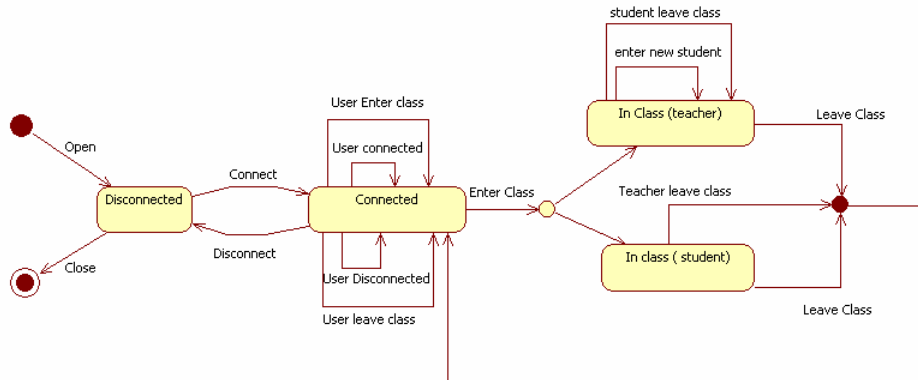


FIG. 2 State diagram of the application where are showed the three states of the client application and the transitions that make pass between them. Note that all the users are handled in the same way until the class begins (the only difference are the users' notifications that every one receives) and that the students must leave the class when the teacher leaves it

The next plug-ins have been implemented to test the system and help in the teaching task

- **Transmission of audio**

Audio handling is one of the most important features of the application. It allows the students to listen to the explanations of teachers and to participate actively in class by answering questions or doing his/her own questions, always under the supervision of the teacher.

When a lecture begins the audio plug-in located in the teacher side begins to sent audio to the student connect to the class. These students can listen the teacher but cannot talk to him. To provide an interactive communication between the teacher and the remote students the first one can grant privileges to the students to talk. In the same way the students can request the attention of the teacher for talk in class by using a virtual way of "raise hand".

- **Chat**

The previously shown plug-ins have not been designed to allow a global communication between the users in the same classroom, only between teacher and a student and vice versa. The chat plug-in (FIG. 3) allows to send text messages to all of the participants of a classroom or in private, between two users (student, teacher). Since the audio is the main way of communication in a classroom, is possible to let the students to use openly this mechanism of communication because they cannot interrupt the class. Nevertheless, the teacher has allays the possibility to restrict the access to the chat to those students when he/she considers needs to be able to communicate.

- **Blackboard**

The most traditional mechanism for teaching has also a virtual equivalent in our application. This plug-in consists basically of a distributed application for drawing. The teacher can draw on a blackboard that is shared by the rest of participants of the classroom.



FIG. 3 Interface of the chat plug-in. It shows the writing area and a log of the received message. The top area is reserved for the list of users.

In order to allow the participation of the students in the class, the teacher has the option to authorize them to also paint on the blackboard.

The plug-in can also store the drawings done in files to let the teacher reuse them later. Also is possible to have several blackboards opened in the same application, by using distinct frames, if the teacher needs more space where to draw.

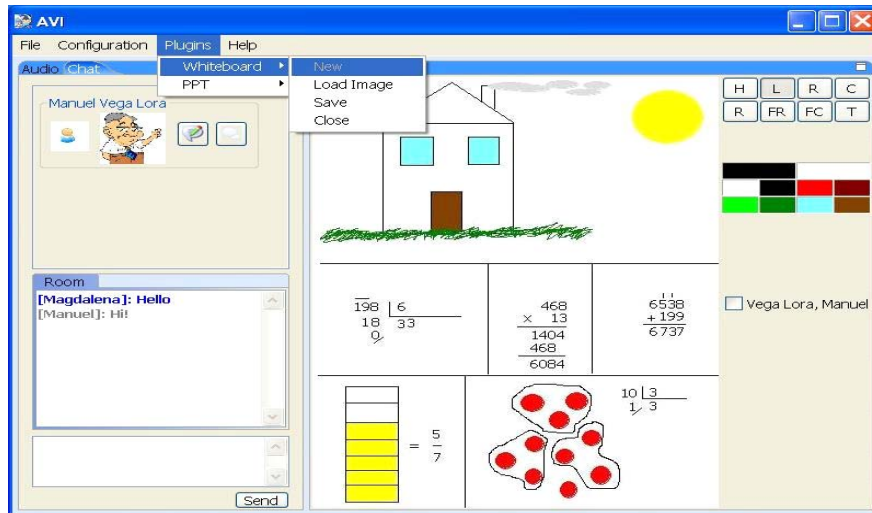


FIG. 4 Interface of the blackboard plug-in for the teachers, where can be seen the drawing tools and the list of users in the class. With this list the teacher can grant privileges to paint to the students. The interface for students is similar to this but without the users list and the drawings tools, which only appear when the student receives grants to paint.

- Presentations

This plug-in broadcasts a presentation from teachers to students. The teacher have full control of the presentation, showing slides forwards, backwards or jump to a particular slide, and every time the professor changes the slide it is sent and shown to the student.

The slides are sent as individual images to the clients. To reduce the delay between two slides, these are sent in advance so the user can have them cached when was needed. The plug-in also notify the clients about the position of the mouse teacher cursor when it is over the presentation. In this way the teacher can point to certain elements of the presentation to help the students to understand the lecture.

- Realization of exercises

Exercises help the teacher to evaluate the progresses made by students. For that reason we have made a plug-in for handling exercises. The plug-in allows the creation of small tests which are distributed between the students together with specific time to be finished.

The teacher can edit the test, by specifying the maximum time to realize it, the number of questions, their values and the types, etc. Currently we support three types of questions: numeric, single choice and multiple choices. The tests can be saved and later restored in disk, allowing the teachers to work in online mode.

One time the test has been seen to the students, they have a limited time to ask the exercises and sent it back to the teacher. When all of them have finished, the teacher received back their answer and a statistics of success for every student

CONCLUSIONS:

In this document a new controlled collaborative teaching environment tool specifically oriented to educational activities is reported. The system is composed by a set of independent client programs and a central server who works as point of contact between the clients.

To simulate the operation of a real classroom the system organizes the users in independent groups, where the teachers maintain the control of the use that the students make of the resources. This organization reduces the number of communication packages interchanged between the clients and distributes the potential network overload that could arise in the teachers' clients.

The students can follow the lecture by an audio system and observe the teacher by using a video camera. They can also participate actively in class answering the questions raised for the teacher or proposing new questions to be discussed. To reproduce this behaviour maintaining the teacher control over the communications, the system includes a method to simulate the "raise hands" of the students and authorization from the teacher.

To extend the application with new functionality, a plug-in mechanism has been incorporated, which allows adding new functions and tools to the central skeleton of the application. As a proof of concept, several light weighted processes has been developed as a set of plug-ins that emulates tools and methods that a teacher or students can use in class, like a blackboard, a chat, presentations slide, etc. Every one of them uses the application as a shell from which obtains the resources needed to execute the work it has been prepared for.

References

1. Xavier Ruiz, Miguel Barzallo, Katherine Chiluiza. 2006. Revista Tecnológica ESPOL (Vol. 19, N. 1, 107-112).
2. GroupLog: <http://www.bath.ac.uk/e-learning/grouplog/>
3. Dougiamas, M., & Taylor, P, 2003. "Moodle: Using Learning Communities to Create an Open Source Course Management System". World Conference on Educational Multimedia, Hypermedia and Telecommunications 2003(1), 171-178. [Online].
4. Swamy, N. Kuljaca, O. Lewis, F.L. "Internet-based educational control systems lab using NetMeeting" Autom. & Robotics Res. Inst., Texas Univ., Arlington, TX; Education, IEEE Transactions, 2002 (Volume: 45, pages: 145-151)
5. L Free . "Review of Windows NetMeeting 3.01" - Electronic Journal of Foreign Language Teaching, 2006 (Volume: 3, pages: 57-63)
6. MSN Messenger: <http://messenger.msn.com>
7. Marratech <http://www.marratech.com/>
8. RELATE <http://www-mice.cs.ucl.ac.uk/multimedia/software/relate-ui/>
9. Pidgin Group <http://www.pidgin.im/>
10. Pidgin-VirtualClassroom <http://developer.pidgin.im/wiki/VirtualClassroom>
11. S. Schnauffer, H. F ußler, M. Transier, and W. Ef-felsberg. Vehicular Ad-Hoc Networks: Single-Hop Broadcast is not enough. In Proc. of WIT2006, pages 49-54, Hamburg, Germany, March 2006, to appear.
12. Hugo Rosales, Ismael Del Águila, Maite Martínez Paradinas and Oswaldo Trelles; "Synchronous distance learning platform: School goes home in Haemophilia"; World Federation of Haemophilia - World Congress - Vancouver 2006
13. Sergio Ramirez, Hugo Rosales, Oswaldo Trelles. "On-line Distance Learning Platform"; 3th International Conference on Web Information Systems and Technologies 2007

Acknowledgments

"This project has been partially financed by the National Institute for Bioinformatics (www.inab.org) a platform of "Genoma España".