COLAB: A PLATFORM DESIGN FOR COLLABORATIVE LEARNING IN VIRTUAL LABORATORIES

M. A. Martínez Carreras, A. F. Gómez-Skarmeta, E. Martínez Graciá and M. Mora Gónzalez
Deparment of Engineering of Information and Communication. 30071 Campus de Espinardo. University of Murcia SPAIN

Abstract: Nowadays collaborative educational platforms are producing an important impact in our society showing this influence in a huge number of researching projects. The use of Information and Communication Technologies (ICT) has gained its prominence in the building of collaborative environments for educational purpose providing new ways of producing learning. So, in order to obtain an educational system which fits with the educational purposes, it is necessary to join pedagogical and technical efforts.

More concretely our developments are focused on Computer Supported Collaborative Learning (CSCL) systems, in which the ground of the knowledge building is the user collaboration. In this paper we are going to deal with the design and implementation of a platform for supporting collaborative learning using tools such as whiteboard, chat, posting tools, modeling experiments tools and other tools which allows view the results from the experiments.

Regarding the design of this platform it is has been used J2EE technologies for supporting the asynchronous communication and Elvin notification server like to support the synchronous communication.

In this paper it will be also described how it has been developed the architecture, showing as well how the tools can exchange events delivered in the communication channel. Finally, we will show an example of collaboration inside this system, and it will be explained how the user can move in this platform as well as how users can coordinate the use of some resources.

Key words: CSCL, Virtual Laboratories, J2EE, notification server

1. INTRODUCTION

Nowadays collaborative educational platforms are producing an important impact in our society showing this influence in a huge number of researching projects. The use of Information and Communication Technologies (ICT) has gained its prominence in the building of collaborative environments providing new ways of producing learning.

Nowadays the University of Murcia (UM) is collaborating as an active member in several educational researching projects. More concretely it is involved in the European Project COLAB[6] "Collaborative Laboratories for Europe" (IST-2000-25035) whose main objective is the development of a CSCL[1,2] educational platform which allows user collaboration in order to solve and simulate some experiments. What is more, in this project the University of Murcia is responsible for defining and developing the architecture to support all the collaboration.

In order to obtain a CSCL system which fits with the educational necessities of teachers and students it is necessary to join technical and pedagogical effort. So, in the development of this project physicist, computer science engineers and educationist are working together to reach a desirable platform.

What is more this system does not pretend to replace traditional education way in fact it is shown as a system used to enrich the learning process encouraging students in the Physic subject through the use of the system.

In this paper we will deal with topics such as how it has been designed the COLAB system and the elements which are included in it, secondly it will be explained the log events caught in the system, after that it will be shown how users can interact with the system and finally it is exposed the conclusions and future work in this project.

2. COLAB SYSTEM

2.1 Requirements

Following the pedagogical instructions all information in COLAB will be managed following this model:

1. All the users will be associated in groups.

2. Each group can access to some “Buildings”. This term is introduced to group different phenomena (experiment simulators).
3. Each building consists of some “Floors”, with the aim to offer some tools and simulators. Each floor implies different level of difficulty.
4. In each floor there are four “Rooms”: Hall, Meeting, Theory and Laboratory. These rooms have specific meanings and different tools: Hall is the first room where users enter after they have been logged in a specific floor, Meeting is used for the communication, Theory includes all the tools for modeling experiments and Laboratory includes all the tools to manage phenomena, which are running on a remote server, and to show their status in client tools.

2.2 Colab Architecture

With the aim of obtaining an architecture which allows to store all the information in the educational platform the asynchronous part of this architecture is based on J2EE technology (Java 2 Enterprise Edition)[3] using JBOSS[4] server in our development. The main reasons for using this technology are the following:

1. The use of Entity Beans help in data base management.
2. The implementation of three layers model allows the separation between the interface, business methods and the database, facilitating the introduction of new applications in the platform.
3. Allow to develop portable code, so it can be run in any platform.
4. This technology is based on well-known standards which help in the system configuration and management.

However, this technology is not enough to solve synchronous collaboration between tools and users. For that reason, it was studied some technologies in order to obtain a synchronous model to support the synchronous collaboration. In our research we have studied the following technologies: JMS (Java Message Service), JSDT (Java Shared Data Toolkit) y DSTC Elvin[5]. Due to its high performance, efficiency and the good results in previous research [7,8], it has been selected the notification server DSTC Elvin as the core of the synchronous system. So it is responsible for managing all the events produced by the collaboration between different clients or by receiving events from remote laboratories or simulators. By specifying different building, floor and group the system will be able to established different sessions in which the distributed information is not mixed with the information generated in others sessions.

The JBOSS server stores all the COLAB structure, including the visual tools, which allows the collaboration between users and simulators, phenomena, users information, relations of groups, buildings, floors and rooms, as well as session files.

In the figure 1 is shown the general schema of the COLAB architecture. So all clients will establish a connection with the J2EE server in order to be introduced in the COLAB structure by selecting group, building and floor. In addition it will be established another connection with the Elvin US. Through this connection a client tool will be able to send and receive events to or from other clients or phenomena (sending events to start, stop the running of an experiment or receiving events with the results obtained in the experiment).

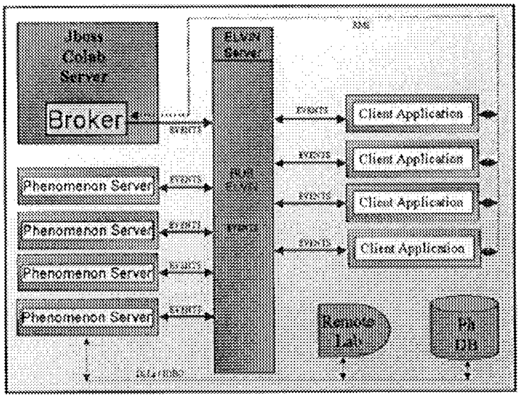


Figure 1. COLAB schema, reflecting the different elements like JBOSS, Elvin server, clients and Phenomenon server (simulators).

All the Visual Tools and the Phenomena are configured using XML files. In the configuration file it will be specified the variables and the values according to the corresponding tool or phenomenon. These xml files are stored in the JBOSS server.

With the aim of facilitating the management of Entity Beans it has been developed a SessionBean called broker which provides methods to create, delete, modify and retrieve all the instances in the database. Each client will establish a connection with a different Broker instance and using them, each client can retrieve all the information inside the JBOSS database.

In order to facilitate the development of new collaborative tools in the platform, it has been developed an API which allows to implement tools and phenomena avoiding to be conscious of how all the elements are related in the system. Therefore, this API offers all the necessary interfaces to develop all the elements in the platform.

The Elvin Bus will be responsible for delivering information between tools, i.e. catching the data delivered by phenomena, allowing subscription to some parameters (in this way tools like graphs and tables can show the changes) as well as communication between users using the chat or the whiteboard. With the aim of helping in the data retrieval and transmission through the Elvin bus, it has been implemented the class "Environment". This entity will be responsible for delivering and retrieving information to the subscribed tools avoiding them to work directly with Elvin.

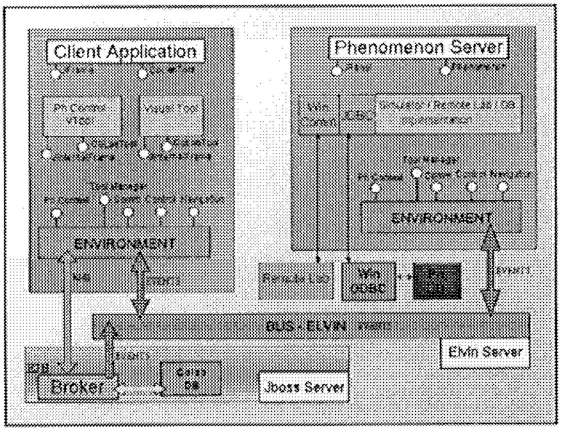


Figure 2. Environment functionality

What is more, all the programmed tools follows the same API offering a common set of methods by implementing an interface called "ColabTool". In this way, the "Environment" can communicate some events to tools invoking some methods from the interface. In order to run the tool in a dynamic way the system will execute the run method in each tool passing the "Environment" and the xml configuration for this tool. So tools will be able to use methods in the "Environment" to pass events to Elvin.

Similar to tools, phenomena will implement the interface "Phenomenon" and in this way they will manage a set of common methods which facilitate the communication through the "Environment".

2.3 Synchronous communication features

In order to maintain synchronous communication it is necessary to take into account the following features[9]:

1. Users in the same session manage the same information (Workspace Awareness)
2. All the resources in the system has to be accessed in a coordinate way which implies concurrency policies.
3. The user connection in the session will be possible at different points in the time. (Late Coming).
4. All the users need to be conscious of online users in their same session. (Presence Awareness)

To solve the first aspect all client tools are subscribed for receiving some events inside a session depending on the kind of tool, so all the users in a session can view the same information.

About the second point, it has been introduced the term of "leader" inside each room. So the first user who enters in a room will be the leader and the responsible for directing the activity progress in this room. This role can change during the collaboration because others users can ask about the control of the session. According to these requests the actual leader can give the control to anyone of them or well ignore these requests.

Furthermore, it is necessary to define a mechanism to establish critical sections which should be executed by only one user in order to obtain a coherent control in the change of some parameters. This mechanism will allow to synchronize the use of some sections and it will be modeled using a Token implementation, represented in our system like an EntityBean with grab and release methods.

So as to solve the late coming of users in rooms the leader will be the responsible for sending this information when he notices a new user has come into the room.

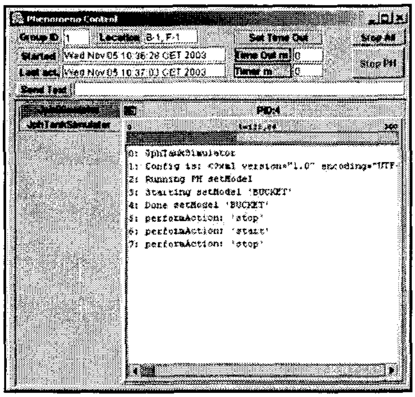
To cover the Presence Awareness feature, the application shows the members list in the left, lighting the online users and showing their location (room) in the session.

2.4 Dynamic phenomena management

In the COLAB system it can be held remote laboratories (hardware entities) and simulators (software entities). Both are represented like "Phenomenon" objects in the system, but they have different performance.

In the first case it would be required that only one group can manage it at a specific moment. In order to facilitate it, a reservation management has been designed indicating parameters such as date, hour and group to book the use of this phenomenon. On the other hand, it has been defined that simulators can be accessed by different groups at the same time, managing different information according to the group, building and floor where the session is held.

With the aim of facilitating the management of the phenomena servers, it has been built an application which is responsible for running and stop any experiment server for each actual session in a dynamic way. So, this tool is listening all the events of starting session and ending session from Elvin in order to run or stop the corresponding phenomena automatically. It also allows sending special messages to some floor, to stop and start specific phenomena as well as showing the progress of any running phenomenon in the system.



Group ID: 1 Location: B-1, F-1
Started: Wed Nov 05 10:36:28 CET 2003
Last act.: Wed Nov 05 10:37:03 CET 2003
Time Out m: 0
Timer m: 0

```
0: JphTankSimulator
1: Config is: <?xml version="1.0" encoding="UTF-
2: Running PH setModel
3: Starting setModel 'BUCKET'
4: Done setModel 'BUCKET'
5: performAction: 'stop'
6: performAction: 'start'
7: performAction: 'stop'
```

Figure 3. Phenomenon Control for the dynamic phenomena management

2.5 Resources Manager

In order to manage and relate all the information in the system it has been built an application which allows, among others tasks, to copy all the information in files as well as restore these files. This tool also provides ways to retrieve the XML information from tools status in a specific session as well as copy from one group to others so as long as they are in the same floor.

What is more, this tool help in the configuration of each "collaborative area" (building, floor, room) setting the necessary tools in the floor according to the purpose of this floor. In this way, the COLAB infrastructure provides several tools, and they can be assigned to different rooms in different floors.

Following the same pattern used for tools, it is possible to assign different phenomena to different floors according to the floor purpose. Both of these aspects give to the COLAB platform adaptability to needs of teachers and students.

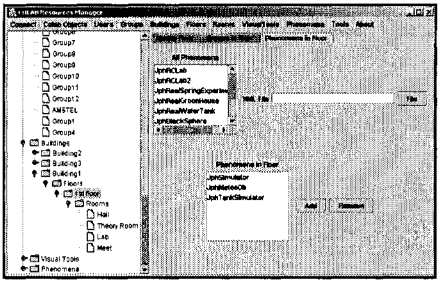


Figure 4. Resources Manager for managing all the data in COLAB

2.6 Colab Client

With the aim of granting that all users work with the same version in COLAB, the client design has been done using the Java Web Start technology. In this way an user will connect through a web page and by clicking a link the "Java Web Start" will be responsible for download the necessary jars for running the application and launch it. If the local machine have the latest jars the Java Web Starts will only launch the application.

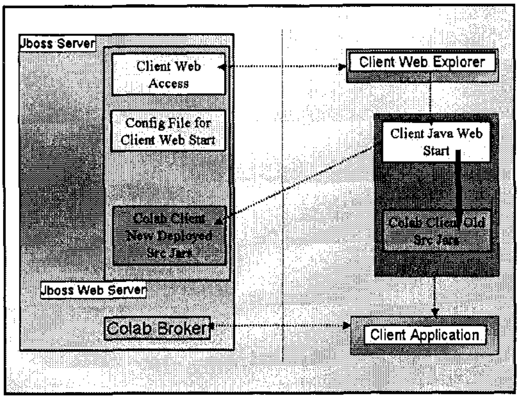


Figure 5. Client connection to Colab

2.7 Collaborative tools inside the system

One of the basis which help in the collaborative knowledge building are the tools used by the group for sharing the experiences of each participant. In this way the results from these interactions are the ground for the knowledge building process[2].

With the purpose of covering the previous aspect it has been implemented the following tools:

a) Chat.- This tool allow the textual communication between users and it is assigned to all rooms in order to facilitate this kind of communication wherever users are working. So users have the possibility of sending messages in a specific room or in all the rooms in the floor.

b) Whiteboard.- This is a tool for the graphical communication. It allows users to use different ways to represent their knowledge, using for example conceptual map to engage different items or to classify any concept.

c) Process Coordinator.- This posting tool help users to follow some goals added by teachers or by users in the same group.

d) Model Editor.- This graphical tool allows design simulations introducing different kind of elements and configuring them with values or parameters according to the group decision. It also provides options to execute this simulation and view the results through the graphics and table tools.

There are others tools which help in the learning process and are the basis for the discussion and the knowledge building of the group, such as simulators, visual tools, graph and table tools and html viewer, which give general concept to students for starting to work in the collaborative area.

In order to continue working with data resulted from previous actions with other tools, there is a repository tool which stores the necessary objects to use them in other activities. In this way is possible to show values generated in previous simulations in the graph or table tools.

3. LOG SYSTEM

In any CSCL system is really important to obtain the traces of the activity progress of users, their participation in the collaborative session and what are the concepts users has been dealing with.

In order to cover these analysis it is saved some events according to the pedagogical instruction. So each important action in each tool is caught and stored in a file. Due to the fact that tools have different behavior, different events will be saved in the different user/tool interaction. In the following figure it is shown some tools and special events generated in the interaction with the whole system.

Table 1. Some action events

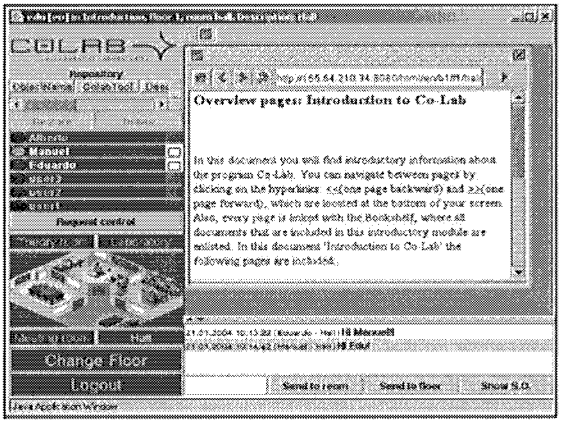
Tool	Action
PC	Addgoal AddSubgoal ShowDescription ShowLinks ShowHints ShowNotes ShowHistory ShowReport ShowGoalTree CompleteGoal GoalName AddNote
General	Tools opening/closing LoginFloor#Building# LogoutFloor#Building# LeaveRoomMeeting LeaveRoomHall LeaveRoomTheory LeaveRoomLab EnterRoomHall EnterRoomMeeting

Tool	Action
	EnterRoomTheory
	EnterRoomLab
WhiteBoard	DrawShape
	MoveShape
	ResizeShape
	ChangeColor
	EditTextColor
ModelEditor	SelectOpenModel
	AddRelation
	DeleteRelation
	AddFlow
	DeleteFlow
	CheckSpecification
	RunModel
	OneStepRun
	StopModel
	ResetModel
Chat	Sendmsg
	SendToRoom
	SentToFloor

4. EXAMPLE OF COLLABORATION WITH COLAB

Previously to a client connection, Elvin, Jboss and the Phenomena control must be running in the server. Firstly the user has to introduce his login and password and as soon as this data is validated the user can select the group, the building and floor where the session will be held. There is another way to connect by selecting an online user of the same group.

After that, the user will be located automatically in the "Hall" room. Once the user is in this room he can move to another room by clicking in the name (Hall, Meeting, Theory and Laboratory) shown in the left part in the application (see Figure 6). The user will be conscious of where he is because the corresponding room image will be light.



Overview pages: Introduction to Co-Lab

In this document you will find introductory information about the program Co-Lab. You can navigate between pages by clicking on the hyperlinks: <<(one page backward) and >>(one page forward), which are located at the bottom of your screen. Also, every page is linked with the Bookshelf, where all documents that are included in this introductory module are enlisted. In this document 'Introduction to Co-Lab' the following pages are included.

Figure 6. Hall Room

As it shown in the Figure 6 in the left part it appears the online users of this group, showing in which room they are located with an icon on the right of the name. A green circles marks who is the leader in a room, i.e. the person who controls the activity in this room. On the other hand, a red circle indicates users who are connected in the room and they are not leaders. Whenever one of these users want to became the leader he can ask for the control using the "Request control" button.

In the upper section of the application it is shown the tool bar and moving the mouse over an icon it a brief description of the tool is shown.

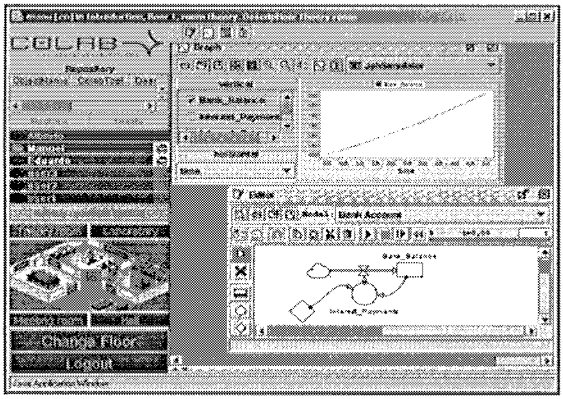


Figure 7. Theory Room

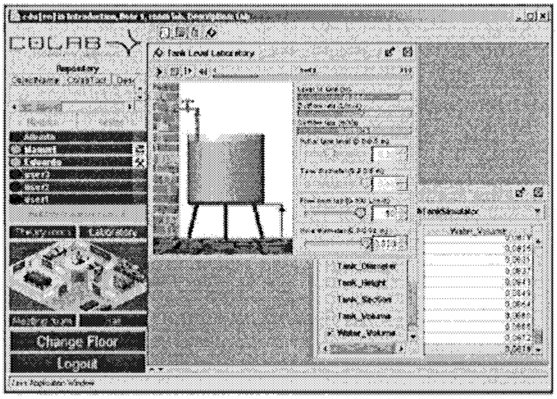


Figure 8. Laboratory Room

In the Figure 7 is shown the Theory room with the Model Editor to model and simulate experiments. In the Figure 8 is shown the Laboratory room with the JvtTankSimulator tool which caught the events from the corresponding phenomenon (TankSimulator) and shows them graphically.

Once the last user leaves the application all the work will be saved in the corresponding session of his groups, and so all the status of the tools can be retrieved in sessions subsequent.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have shown how it has been developed a platform for the collaborative learning by using virtual laboratories. So this platform held different collaboration areas where users can share their experiences and build new concepts or models by using tools such as whiteboard, chat, process coordinator and model editors. In this area the users are provided with several tools for obtaining experiments results, which will be the ground for the collaboration.

As a result of this architecture design and its components, it has been build an architecture which provides the "*basic ingredients for collaborative discovery: frames of reference, collaborative tools, and experimentation space"[2].*

In the development of this tool it has been built an architecture based on application servers in which the integration of new tools could be performed easily. In addition, these tools could be assigned to different floors with different XML files, according to the use of this floor.

The system configuration is based on standards widely used such as XML, facilitating the configuration of each application using this standard.

Furthermore, this platform and its set of tools are shown to the final user like an application easy to manage, with intuitive graphical tools which can be managed in a friendly way for any kind of users. In this way we pretend to encourage users to learn Physic by the use of this system.

What is more it has been obtained a platform which is portable to any collaborative area. These features are obtained due to the fact that our development is based on the J2EE standard. In addition, the synchronous schema is valid for any kind of application because the subscriptions are performed by the tools and they are managed by the system using the "Environment" interface. For that reason, the whole design could be used in a different areas, such as working task, and other educational purposes.

In addition this system provides a way of catch and save important events from the collaboration in the system, and in this way teachers will be able to trace them by the use of tools which will be developed in next phases of the project.

ACKNOLEDGEMENTS

The development and researching shown in this paper has been financed by the European project *Collaborative Laboratories for Europe* IST-2000-25035.

REFERENCES

1. Stahl G. "Contributions to a Theoretical Framework for CSCL". In CSCL 2002.
2. Joolingen, W.R. van (2000). "Designing for collaborative discovery learning". In G. Gauthier, C. Frasson and K. VanLehn (Eds). Intelligent Tutoring systems. (pp. 202-211).Berlin: Springer.
3. Link J2EE http://java.sun.com/j2ee/
4. Link JBoss http://www.jboss.org/index.html
5. Link Elvin DSTC Elvin http://elvin.dstc.edu.au/
6. Link Colab http://colab.edte.utwente.nl/
7. Leinonen T., Hakkarainen K., Appelt W., Dean P., Gómez-Skarmeta A., Ligorio B., Lipponen L., Merisaari L., Mielonen S., Pontecorvo C., Sligte H., & Vosniadou S. (2001) "ITCOLE Project: Designing innovative technology for collaborative learning and

knowledge building". Proceedings of the World Conference on Educational Multimedia, Hypermedia & Telecommunications. Tampere. Finland, June 25-30, 2001.

8. Pedro García López, Antonio Gómez Skarmeta, Robert Rallo Molla. "ANTS: a new Collaborative Learning Framework". ". Proc. Of the European CSCL. Maastricht (Holanda) Marzo 2001.
9. Begole J. "Flexible Collaboration Transparency: Support Worker Independence in Replicated Application-Sharing Systems". Phd.1998.
10. M.A. Martínez Carreras, A. F. Gómez-Skarmeta, E. Martínez Gracia, M. Mora Gonzalez. "Colab: A Platform for simulation in collaborative environements in virtual laboratories". Proc. Of mICTE2003 (pp. 1591-1595). Badajoz (Spain)2003
11. Link Colab http://colab.edte.utwente.nl/