

# CLASS: A Computer Platform for the Development of Education's Collaborative Applications

**Luis A. Guerrero**

Computer Science Dept.  
Pontificia Universidad Católica de Chile  
*luguerre@ing.puc.cl*

**David A. Fuller**

Computer Science Dept.  
Pontificia Universidad Católica de Chile  
*dfuller@ing.puc.cl*

## Abstract

In spite of the important development that the Web has gone through in the last years, the field of education has not taken fully advantage of this development. Using this technology on the educational process could bring different benefits for all its users. In this paper we propose a computer platform on the Web for the development of education's collaborative applications. This platform includes access control through passwords, and queries to a database with structured information, semi-structured data and meta-information. We also provide some facilities of collaboration for the applications developed. Finally we propose some applications that can be supported by this platform, and we provide the possibility to built other applications not only on the Web.

## 1. Introduction

The development of the World Wide Web in the last years has been so amazing, that today the meaning of Web and Internet is almost the same, mainly due to the addition of other tools of Internet on the Web's browsers, like e-mail, ftp, news, etc. This development has become very interesting on the collaborative applications due to the fact that the Web has been successfully introduced in many educational institutions, organizations, companies and homes. The Web has entered into places where the people live, study and work.

In the educational field the Web is useful in the research area, since almost all the required information can be found on the net. However, the basic teaching model follows similar patterns than the traditional models of teaching used since a long time ago. We consider that the educational area has not taken advantages of the Web development, mainly due to the lack of tools that support this work, especially tools that support the collaboration between the users involved in the course.

The CLASS project ("Collaborative Learning Applications for Students' Support") is a platform for the construction of collaborative tools on the Web that supports the educational process at the university courses. In this paper we describe the CLASS platform, previous work, the applications that CLASS can support and the facilities for the development of collaborative applications.

Currently there are a few projects that have similar goals. The Habanero project at NCSA [w2] is a framework for sharing Java objects with colleagues distributed around the Internet. However, it does not provide tools to manage data or facilities to the development of Web applications. BSCW ("Basic Support for Cooperative Work") [w3] enables collaboration over the Web. BSCW is a 'shared workspace' system which supports documents upload, event notification, group management and little data awareness. However, it does not provide tools to manage data or facilities to the development of applications. Web-CT ("World Wide Web Course Tool") [Goldberg96] is an easy-to-use environment for creating WWW-based courses that are otherwise beyond the ability of the

non-computer programmer. However, it does not provide collaboration facilities and data administration.

CLASS contains the facilities of the above three systems, sharing objects through a Web interface, with data administration tools and facilities for the construction of educational collaborative applications.

## 2. The Course Memory

During the development of a course we can store (for example in a Web document) the course text, tests and homework of the previous period, links to other related data and any other type of information that can be useful for the course. We call all this information the *course memory* [Conklin92], and is of interest for students, assistants and the lecturers. For example, students can use this repository to study and assistants and the lecturer can use it to prepare the lecturers and exams.

This form of keeping the course memory stored in an HTML document is very common and is used by many courses at different universities [w1]. However, there are many problems when using Web pages to store the course memory. First, the questions and answers arose in class, comments and observations made by the lecturer or by any student, and all the non-written information or informal information is lost. It is possible that this information can be required the next class period, but because it never was stored, the new students cannot use it.

A second problem is that they always

show the same view for every type of users, i.e. students, assistants and the lecturer look at the same information even when their needs are different. A third problem is the lack of data awareness. In order to check if some information has been added, deleted or changed, a user has to read the entire document carefully to check for differences. Obviously, this is not an easy job when the document is very extend or composed of many Web pages.

Figure 2.1 shows an OMT objects diagram [Rumbaugh91], with the three major components of a course. The *course* has a name, reference number, section number, schedule and date. It also has *administrative information* with homework rules, exams dates, bibliography, objectives and others. It also has *participants*, which is the information regarding all the participants of the course, especially the lecturer, assistants and students, and the access rights of each type of user. Finally we have the *course memory*, containing the subjects of the course, information of previous courses (homework, exams, etc.) and the information of the actual semester (questions/answers, news, comments, etc.).

Collaborative applications that support the teaching-learning process should consider a course memory that captures not only the formal but also the informal objects, provide data awareness and satisfy all the different needs of the different participants. With collaborative applications supporting the teaching-learning process, that organizes and manages the course memory efficiently, and also stimulate collaborative learning, we can expect better learning and teaching processes,

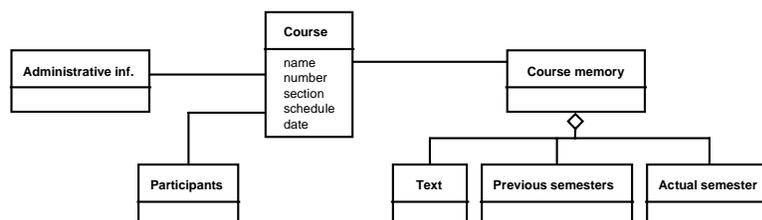
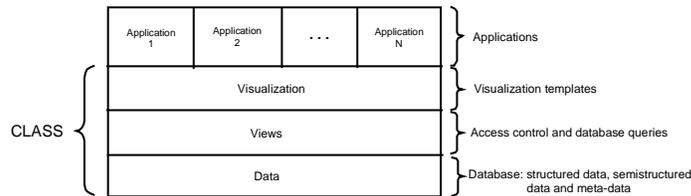


Figure 2.1. Structure of a course.



**Figure 3.1.** CLASS Platform to support other applications.

a better performance in group's work and, in general, a positive attitude towards the course.

We already designed and implemented a system that allowed us to experiment in the collaborative learning processes. We called this system, CollaboratiWeb or CW [Castillo96]. CW is an application that allows, using Web pages, identify users through a password, assign access rights and provide a simple data awareness. It also organizes the course memory through the storage of ideas, comments and queries. Also, it allows the entire or partial view of Web pages, according to the users' access rights. The complete course text is storage on a Web document. All this was implemented by the introduction of CGIs on the Web server.

### 3. The CLASS System

CollaboratiWeb had more success than we expected. However, after one year of use, we observed that it could be redesigned in many aspects. For example, in order to provide more flexibility in the data administration and data queries, we need more than simple HTML documents (like in CW). Also, we need more facilities for the application construction process and for the communication and collaboration between users. For these reasons, we designed and created the new system CLASS. This paper reports this new design.

As is shown in the figure 3.1, CLASS is a support platform for the development of learning collaborative applications.

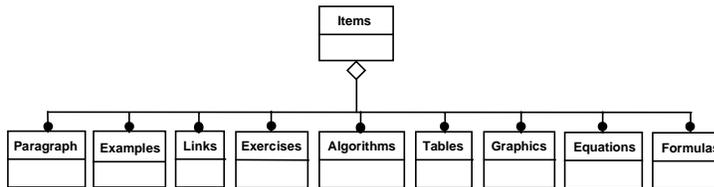
At the lower level, CLASS stores data as multimedia didactic objects and collaborative objects. This is composed by structured and semi-structured information. Over these objects there is a view and query layer. This layer takes a query from the upper level, checks the access rights and brings back the objects that match this query. The visualization layer takes these objects and adds the needed tags for the proper presentation to the user. Finally the applications take this information and present it to the user by their own interface. The applications have the option of modifying the final view of the data. Some of the main aspects of the CLASS design are detailed below.

#### 3.1 Structured and semi-structured data

In order to provide a better query system for the course memory, it is necessary to organize the data according to a data administration policy. We need more flexibility on the information management processes than what we can have from Web pages.

In general, HTML documents consist in semistructured data. *Semistructured data* is data that has some structure, but it may be irregular or incomplete, and not necessarily conform to a fixed schema [Chawathe97].

The text of the course is similar to an HTML document, since it consists of items, that at the meantime consist of multimedia objects such as graphics, examples, algorithms, paragraphs, videos, sounds, links, formulas, exercises, tables and equations, as shown in figure 3.2.



**Figure 3.2.** Objects that compose the items

The above structure of objects is best represented as semi-structured data [Chawathe97], and not as structured data, because these objects can be composed of any possible combination. Thus, we need a semi-structured data manager and not a traditional database manager.

Due to this problem, to represent and manage the semi-structured information in CLASS, the OEM model (“Object Exchange Model”) designed at the Stanford University is being used [Papakonstantinou95]. For querying over data, the semi-structured query languages Lorel and Chorel [McHugh97] also designed at Stanford University are being used. The query results are generated to an ASCII file that can easily be converted into HTML documents or into any other format.

The course information also contains structured data as user tables, administrative information, and others. This type of information is handled by a traditional DBA. The management of the semi-structured and structured data is encapsulated inside the CLASS data layer. The views layer only asks for objects, and the way of how the information is managed is transparent to this layer.

### 3.2 Roles and Views

As mentioned before, we consider three types of roles: the *lecturer*, the *assistants* and the *students*. Each of these participants has different needs. There is a fourth role which is the *system manager*, who creates the users’ profiles, assigns and changes passwords, and other administrative tasks.

For each object stored on the course memory, we have information regarding the access rights of any type of role (basically *create*, *delete*, *modify* and *read*). This is the reason for getting different results from the same query made by different users. The CLASS view layer is in charge of controlling the access to the objects accordingly to the user roles. Once the view layer checks the access rights, it asks for the information to the data layer.

Generally, the views are *dynamics*. This means that a query is made at anytime, and the objects are searched on the course memory. A problem may arise when the number of users is large. In this case the server may not be able to produce the users’ queries with the required performance, becoming a bottleneck. In order to solve this problem, *static views* can be defined. These are preprocessed and stored queries ready to be delivered. For example, an application can define as static the most common views. These static views are refreshed when the course memory changes.

A view can also be stored somewhere else, for example on a diskette or on a hard disk. These views are *local views*, and over them the system does not provide awareness information (see next section) since they will run only on top of a client’s Web browser, i.e. with no connection to a Web server. The non-local views are *connected views*, which mean that a Web and a CLASS server support them. Over them, CLASS does provide awareness information.

Thus, the four types of defined views in CLASS are static, dynamics, locals, and connected. The type of view required depends on each application developed on CLASS. We describe specific examples of views when we refer to some of the applications supported by CLASS.

### 3.3 Awareness

The awareness is a general view of the shared workspace on collaborative applications. It includes aspects such as knowledge of people that are around, what are they doing or have done recently, who are they taking with, and any other information that helps to answer the questions: who?, what?, how?, when?, where? and why?. One of the most acceptable awareness definitions is: “an understanding of the activities of others, which provide a context for your own activity. This context is used to ensure that individual contributions are relevant to the group’s activity as a whole, and to evaluate individual actions with respect to group goals and progress” [Dourish92].

In order to make a better explanation about the way in which this concept is incorporated into CLASS, we define the awareness in two types: data awareness and user awareness. Data awareness is referred to the meta-information or information about information. It is information about the course memory that can be useful to each user that works in some application. CLASS manages awareness information for each user in the system, keeping files with statistics object use. Data awareness is available on the repository and each application can use it.

For example, a student can be interested in knowing the items of the subject he has studied and which has not. He may also be interested in knowing if any section has been modified, deleted or added to the text. This information is available and an application can ask CLASS to show it to

the user. Note that data awareness may well be different for each role, so the application must provide it in the proper way.

User awareness refers to those participants who are using the application and what are they doing or what have they done recently. For user awareness, CLASS keeps information about connected users as well as a log file of the users’ actions. Again it is the decision of each application to present or not this user awareness information, and the way of presenting it. Each application should resolve the privacy problem. Regarding this, CLASS offers the option that a user becomes a “ghost”, where CLASS will not record its activities. This option is available for the applications to consider.

### 3.4 Collaboration

CLASS supports the collaboration on the applications through both schemes: *collaboration facilities* and *collaboration objects*.

Some of the facilities for collaborative applications that we mentioned already are role’s definition, access rights over objects, and data and user awareness. Other type of facility is the communication between users. The audio and video support is designed inside CLASS, but its implementation will be left for future work. On other hand, a collaborative object called mailbox supports the asynchronous information. It is explain below.

CLASS manages the collaboration objects and the objects on the repository in a similar way. These objects have constructors and methods that can be used by the applications. One of the main collaborative objects is the *box*. A box is a repository where any type of information can be stored. It has a constructor and a destructor, and methods to put objects in,

put them out, see and list them. Also the box can be opened, closed, saved and loaded. Likewise, it also permits to give expiration date and time for the objects pushed into the box. It also has assigned users, with rights over the contained objects. A box is invisible for the users that are not assigned to it.

We can create another type of object named *session* from the box. A session has a beginning and ending date and time, has an owner (who creates the object) and has guest users. This session has a box to store the information produced during the session. This box has assigned all the users that belong to the session. The user's operations over the objects in the box are defined by the owner or coordinator of the session and will depend on each application.

We can build many applications from box and session objects, such as voting systems, discussion systems, news, brainstorming, etc.

The *mailbox* is another object that can be built from the box. A mailbox is a box with an address for each user of the system, where basically messages can be stored (as a matter of fact, any object can be mailed). When a user enters to any of the applications supported by CLASS, the address is checked and the messages are delivered.

Those *messages* sent through mailboxes are another type of collaborative object supported by CLASS. Besides having text, they have the number of times the users will see it and an expiration date and time. Box provides this expiration date. These mailed messages can be classified in three types: ordinary *messages*, *notifications* and *reminders*. The notifications are messages that are displayed only once and are generated for other applications not for other users. The reminders are messages that are going to be displayed to the users a

certain amount of times or until an established date and time, for example a message like: "Deadline to received the papers: June, 30<sup>th</sup>". Each application uses the type of message that they need.

*Stick-on* is another collaborative object supported by CLASS. They are labels that can be stuck to any other object where a simple text can be written.

All these collaborative facilities and objects make CLASS a very powerful platform for the construction of asynchronous collaborative applications, implemented in a client-server architecture. A client can place a requirement into a box. Then a server can take the requirement, process it and place the answer in the box, sending a notification to the client.

#### 4. Visualization

The *visualization* is the last layer in the CLASS architecture, and it defines the way in which the different objects are shown to the user. There is a set of templates that define the presentation rules of the objects in the browser. The application developer can build templates to display objects on a specific browser.

Even though the CLASS architecture design is open, the main applications were designed for a Web browser. Due to this, the visualization process of the objects in our examples consists of HTML tags put by this layer. Thus, each application receives from the visualization layer, HTML documents ready to be presented to the browser.

This visualization layer gives to the applications the user requested objects, the visualization tags and the presentation order. It includes the tags and images for awareness, such as different color bullets to indicate changes or new items.

The problem of the awareness information visualization is complex and very dependable on each application. Looking for a better way to provide awareness visualization to the applications supported by CLASS is a future research area.

## 5. Applications Supported by CLASS

CLASS offers a model for manage and query the multimedia objects of a course, access controls using passwords, collaborative objects, collaborative facilities, visualization and awareness information. Through these services we can build applications that support the user tasks involved in the learning-teaching process.

For example, let us assume that we want to build a simple lessons' browser for a lecturer to present his lessons using a Web browser. First we should allow the lecturer to select the subject covered in the lesson. In the platform there is an object called *didactic objects selection matrix* which looks in the data layer for the number of available chapters in the course and for the kind of objects on each chapter. The visualization of that is shown in the figure 5.1.

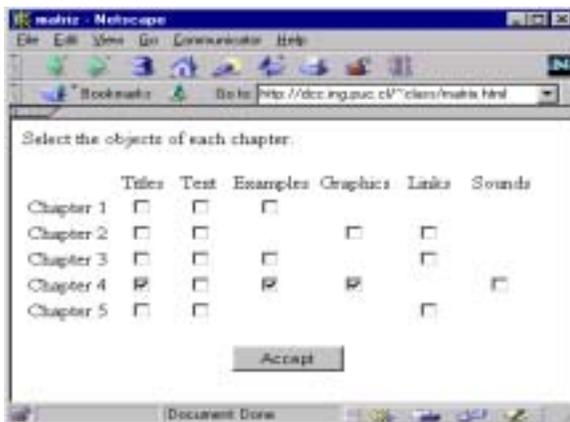


Figure 5.1. Selection matrix of objects by chapter

In this case, the lecturer is selecting for his lesson the titles, examples and graphics that are present in chapter 4. Once he

selects the needed information, he should be provided with the navigation tools that allow him to present the lesson. In this case, the application developer can use an object called *index*. This object can skip directly to a predetermined place or to another Web document. The visualization of this object plus some buttons for the navigation is shown in figure 5.2 for the generated lesson of chapter 4.



Figure 5.2 Lessons browser.

This browser allows the lecturer to go forward in a predetermined sequence or skip directly to an index sheet. A button allows hiding the index in order to expand the space of visualized text. Thus, the collaborative applications developer uses the objects defined in the platform as building blocks, together with other objects defined by him.

The platform is composed by a set of scripts (JavaScript), applets (Java) and CGI's (Perl and C). The programmer must combine all of them to create an application.

Others applications actually designed and supported by CLASS are:

### 5.1 Administrative Module

This module more than an application is part of CLASS, because is here where the users, passwords, roles and right accesses are defined. Also here the maintenance of the information takes place; for example:

create, delete, modify and others. Only the system administrator can use this module.

## 5.2 Lessons Editor

Through this application the lecturer can review the lectures to be taught. He is able to decide the items that will be covered in class. He can save the lecture on a diskette, on a net directory or on the hard disk. The application offers a navigator system to show the class to the auditorium like slides.

To this application, CLASS offers the query system by items and the creation of a Web document by which the lecturer will navigate to give the lecture. The lessons editor allows generating a static and local view that can be carried and shown in any computer with a HTML browser, even when this computer is not connected to the net.

## 5.3 Distributed Synchronous Tutorials

Through this application the lecturer or any of his assistants can make tutorial sessions to a group of students at a set time and from any place, for example from their house at 10:00pm. The tutorial coordinator opens the session at a set time and the students' computers are synchronized to see the same screen as the coordinator.

CLASS provides the queries to the repository and the navigation system. CLASS also offers facilities to built (using a *box*) a discussion or voting system, for example to set the time of the tutorial session. Also provides the *session object*, which helps the tutorial coordinator to define the schedules and the participants.

One of the main problems is the synchronous feature of the application, for example for the audio and video transmission. These aspects are not yet well supported by CLASS and this is an area of future research.

## 5.4 Coursework Support

With this application the students can make their coursework in a collaborative way (asynchronous and distributed). Groups of students are able to do their shared work in an asynchronous and distributed fashion, using this application that provide collaborative features. It is basically a collaborative editor with versions, protected text, messages and stick-ons.

For the development of this application, CLASS offers the box, where all the work made by each member of the group can be stored. Besides this, CLASS has the stick-on object that can be used in order to make comments to the work made by other members. CLASS also provides a communication system using mailboxes.

## 5.5 Students Views

Through this application the students can navigate through the items of the course, in a sequential way (as reading a book), like in a hyperdocument or through queries or predefined views of the repository information, for example: "all the exercises about threads".

This application uses the system view that CLASS offers, and it can be supported by data awareness in order to indicate modifying or new sections, or indicate sections that has not been studied yet.

## 5.6 Collaborative Construction of Didactic and Multimedia Material

The idea of this application is to provide a collaborative environment for lecturers and assistants to produce and edit the material for the lectures in a distributed and asynchronous way. Paragraphs, graphics, video, audio, algorithms, examples, links, tables, equations and formulas compose the items.

CLASS offers facilities to define the work sessions (*session object*) and the share information's repositories (*box object*) for the items. CLASS also offers the object delivery system (*mailboxes*).

### 5.7. Other Applications

In CLASS we can easily build collaborative applications that can be used as part of other more complex applications. For example, we can create a news system using a box, a voting system, brainstorming systems, pre-meeting systems and many other applications based on collaborative objects and facilities, access right controls and roles definition, awareness information and the object storage and queries facilities. We expect to design and implement many more applications on top of CLASS.

## 6. Conclusions and Future Work

In this paper, we presented an object-oriented platform to support the construction of collaborative multimedia applications for the teaching/learning process, running on top of a Web browser. The facilities provided by CLASS allow us to take major advantage of the Web. As an example we showed a simple lesson's browser with dynamic behavior, i.e. the application will produce a lesson according to the user's needs.

Most parts of the platform are currently implemented, and some applications are already being used. However, the platform will be more stable only after having more experience with the construction and use of these kind of collaborative applications.

Audio, video and synchronous communication support is a future research area as well as a better support and visualization of the awareness information. There is also a lot to do regarding the design and construction of tools and applications over the CLASS platform.

For us, it is extremely important to experiment with the applications, measuring their impacts on the teaching/learning processes. This provides us with the required feedback to improve our hypotheses.

We still have to work on the understanding of a collaborative applications generator. Specifically, it is our intention to provide an application on top of our platform that supports the construction of collaborative applications by using visual programming.

## Acknowledgments

This work was partially supported by the Chilean Science and Technology Fund (FONDECYT), grant 195-0880. Special acknowledge is granted to Prof. Jennifer Widom and Mr. Jason McHugh at the Stanford University Database Group for allowing us to use their LORE system.

## References

- [Beaudouin92] Beaudouin M. and Karsenty A. *Transparency and Awareness in a Real-time Groupware System*. Proceedings of the UIST'92, pp. 171-180. USA, 1992.
- [Castillo96] Castillo M.A. and Galarce P. *CollaboratiWeb: Apoyo Tecnológico a Docencia*. Proceedings of the VI Jornadas Chilenas de Computación. Valdivia, Chile, 1996.
- [Conklin92] Conklin J. *Capturing Organizational Memory*. Groupware 92, pp. 133-137. San Mateo, CA, USA, 1992.
- [Chawathe97] Chawathe S., Abiteboul S. and Widom J. *Representing and Querying Changes in Semistructured Data*. Technical report, Stanford University Database Group. USA, 1997.
- [Dourish92] Dourish P. and Bellotti V. *Awareness and Coordination in Shared*

*Workspaces*. Proceedings of CSCW'92, pp. 107-114. Canada, 1992.

[Dourish92b] Dourish P. and Bly S. *Portholes: Supporting Awareness in a Distributed Work Group*. Proceedings of CHI'92, pp. 541-547. USA, 1992.

[Goldberg96] Goldberg M., Salari, S. And Swoboda P. World Wide Web – Course tool: An environment for building WWW-based courses. Computer Networks and ISDN Systems, pp. 1219-1231. Proceedings of the 5<sup>th</sup> International World Wide Web Conference, Paris, France, May 1996. Available at URL: [http://www5conf.inria.fr/fich\\_html/papers/P29/Overview.html](http://www5conf.inria.fr/fich_html/papers/P29/Overview.html)

[Guerrero96] Guerrero L., Romero F. and Zegarra J. *Una Herramienta para el Apoyo del Trabajo Colaborativo en Escolares*. Proceedings of the Educational Software Workshop, pp. 183-192. Valdivia, Chile, 1996.

[Hudson96] Hudson S. and Smith I. *Techniques for Addressing Fundamental Privacy and Disruption Tradeoffs in Awareness Support Systems*. Proceedings of CSCW'96, pp. 248-257. Boston, USA, 1996.

[McHugh97] McHugh J., Abiteboul S., Goldman R., Quass D. and Widom J. *Lore: A Database Management System for Semistructured Data*. Technical Report, Stanford University Database Group. USA, 1997.

[Palfreyman96] Palfreyman K., and Rodden T. *A Protocol for User Awareness on the World Wide Web*. Proceedings of CSCW'96, pp 130-139. Boston, USA, 1996.

[Papakonstantinou95] Papakonstantinou Y., García H., and Widom J. *Object Exchange Across Heterogeneous Information Sources*. Proceedings of the International Conference on Data

Engineering, pp. 251-260. Taipei, Taiwan. March, 1995.

[Rumbaugh91] Rumbaugh J. et al. *Object-Oriented Modeling and Design*. Prentice Hall, 1991.

[Sohlenkamp94] Sohlenkamp M. and Chwelos G. *Integrating Communication, Cooperation and Awareness: The DIVA Virtual Office Environment*. Proceedings of CSCW'94, pp. 331-343. Toronto, Canada, 1994.

## Webliography

[w1] <http://www.ing.puc.cl/Escuela/web/cursos/>

[w2] <http://www.ncsa.uiuc.edu/SDG/Software/Habanero/>

[w3] <http://bscw.gmd.de/>