Using a WWW Server to Access and Manage a Gemstone Smalltalk Server Supporting a Virtual Collaborative Learning Organisation

Frédéric HOOGSTOEL
E-mail : Frederic.Hoogstoel@univ-lille1.fr

Grégory BOURGUIN
E-mail : Gregory.Bourguin@univ-lille1.fr

Laboratoire TRIGONE Bâtiment B6
Université des Sciences et Technologies de Lille
59655 VILLENEUVE D'ASCQ
FRANCE.

I. Introduction

The Co-Learn Web Interface is the first step of the integration of a full CSCL organisation support system with the World Wide Web. It allows users to browse and manage the co-operation settings of an Open and Distance Learning Environment according to their roles in the virtual organisation. After the first part which describes the Co-Learn background, the second part sets out the Co-Learn Web Interface and its mechanisms. The last part concludes by showing the limits of our first implementation and presenting the perspectives of extensions of this work.

II. Co-Learn Background.

The Co-Learn environment has been designed to support Computer Supported Collaborative Learning in training institutions. It is a kind of global CSCW environment, like studied by the MOCCA project and described by [Benford 1991]. This means that Co-Learn integrates long-term/short-term and asynchronous/real-time collaboration tools. It allows the management of persistent collaborative environments represented by the Virtual Room metaphor [Madsen 1989][Hammainen, Condon 1991]. The goal of Co-Learn is to support the six issues we have identified for CSCW in organisations [Hoogstoel 1995] : communication, co-ordination, implication, cohesion, organisation and information sharing. The key issue of such an environment is tailorability. The Virtual Room is an elementary unit of the Organisation Space and a macroscopic element of the Activities Space (similarly to a workspace in CoopWWW [Appelt 1996] or a locale in wOrlds [Fitzpatrick, Kaplan, Tolone 1995]. The functional tailorability is assumed by allowing organisation members to instance
and tailor Virtual Rooms. To tailor a Virtual Room, you choose the activities, documents and participants to include and you assign the roles of the participants in the activities. Since managing such a tailorable environment is very complex, the system proposes default configurations that can be easily modified. By offering basic tools and collaboration policies and allowing the creation of new ones, the system aims to constitute a meta-environment supporting the bootstrapping strategy, like envisioned by [Engelbart 1992].

III. The Co-Learn Web Interface.


Strong reasons of choosing the Web as an environment for co-operative work have been broadly discussed. For example, the last ERCIM workshop on CSCW and the Web has recalled that Web supports standardised hypermedia data handling, global address space, large scale and distribution, offers a well-known interface and constitutes an open system technology [Walther 1996] [Dix 1996]. Actually, we can notice a growth of the number of various groupware tools integrated into pages on the WWW. But these tools are usually not integrated in a Co-Learn-like global CSCW environment. The role notion is generally not supported, and when it is, there is no sharing or coherent managing of the user roles between the tools. On the other side, the CoLearn CSCW environment would benefit from its integration in WWW environment because of the WWW advantages exposed before. That's why we are currently porting Co-Learn on WWW. This paper describes the first completed step of this work which concerns the management of the co-operation settings. This new management environment allows users to manage the organisation from any Web Browser as it was only possible before with the Windows Co-Learn administration tool. Moreover this environment can be used as an Animation Assistant [Hoogstoel 1992] during the co-operation sessions in virtual rooms.

2. Functionality Offered To Users.

The management functions can be classified by two aspects of the user profile: his expertise level and his roles. The basic management facilities of Co-Learn allow to tune the organisational and structural context of co-operation. It handles organisational and structural roles, courses, virtual rooms and users. The advanced management facilities of Co-Learn concern the fine-grained set-up of the co-operation inside the virtual rooms: it allows to choose tools, functional roles and collaboration modes. The meta-management facilities of Co-Learn support bootstrapping by defining new subclasses of co-operation objects (rooms, roles, co-operation modes). A set of rules defines permissions of users to do these actions, according to their roles. The Users Administration role is necessary to handle users at the level of the organisation. The Structure Administration role allows to perform the basic and advanced management functions handling the courses and the rooms. A Teacher in a room can perform basic and advanced management functions modifying the objects inside the room.
3. Design and Realisation.

a) Client-Server Communication Model.

The initial requirement was to move as it was the Windows Co-Learn administration tool on WWW. The existing interface was based on Windows forms and the administration tool didn't provide group feedback. Contrary to most of the Co-Learn tools which rely on a peer to peer communication model (which would require to be implemented in a Java-like language), the administration tool relies on a usual client-server communication model well supported by HTTP and the submit mechanism. We chose Perl to write CGI scripts because of its portability and string treatment facilities.

b) Data Handling and Dynamic View Generation.

As described before, the Co-Learn web interface allows User to inspect, create or modify data of the organisation model according to his role in the organisation (see the User Model). This data can be accessed through a set of administration views of the organisation model. Each view type is implemented by a generic HTML page on the WWW server. A view (an instance of a view type) is implemented by a specific HTML page (an instance of a generic HTML page) displayed on the client side. A CGI script works as a factory by instancing generic pages in specific pages containing the required data extracted from the Co-Learn objects base. This extraction is performed by a method call (to a Co-Learn object) which has been created from a request translated by a dynamic interface invocation mechanism. The parameters of this request are the field values of a form (included in a specific page) filled in by a user. Thanks to an authentication mechanism, the Co-Learn object, which will perform the method call, knows the identity of the requester and can adapt his reaction by merging the requester roles and the organisation rules. [Fig. 1] below shows the links between the components and mechanisms described above.

IV. Conclusion & Perspectives:

Figure 1: The Co-Learn WWW gateway communication architecture
The Co-Learn users can now use the Web to configure their co-operation environments. They can also use it as an animation assistant during their co-operation sessions by dynamically setting up the tools, participants and roles involved in the virtual room. This functionality is offered thanks to a Dynamic View Generation mechanism which allows to inspect and to edit the data encapsulated in Smalltalk objects representing the organisation.

A drawback of the actual implementation using HTML forms and CGI scripts is at the interface level. Since we decided not to use frames because they are not included in the last HTML version (and then not supported by all WWW browsers), it was impossible to enhance the interface with Smalltalk-like browsers (which would better merge global and detailed views) in the management pages. Using the Java language should avoid this difficulty and allow to maintain durable connections between the client and the Co-Learn server. This way we could support group feedback.

The system we described here has been implemented to be integrated with a courseware development and publishing system based on HyperG [D'Halluin et al. 1996] to constitute a multimedia distance education global environment on the web, in the framework of the MODEM European project. The Co-Learn side of this global environment will manage the co-operative activities space and the HyperG side will manage the documents space. During the next months, we will enhance this Co-Learn browser to better support the time dimension of the activities space and to allow the management of technical resources (e.g. audio-bridges). In the long term, this tool will allow meta-management by supporting design, development and integration of new co-operative activity types and roles in the Co-Learn object-oriented framework.

V. References


CSCW (ECSCW '95), Stockholm. 1-16.


