

COLLABORATIVE VIRTUAL ENVIRONMENTS: GOING BEYOND VIRTUAL REALITY[®]

Roberta L. Gomes⁺, Guillermo J. Hoyos-Rivera^{*}, Jean-Pierre Courtiat

{rgomes,ghoyos,courtia@laas.fr} LAAS – CNRS, Toulouse, France.

ABSTRACT

With the growing demand for collaboration technologies, several CSCW (Compute-Supported Cooperative Work) systems have been developed. CVEs (Collaborative Virtual Environments) represent an important category of CSCW systems that generally make use of 3D shared space to provide collaboration facilities. However in most CVEs collaboration is restricted to the Virtual Reality context. In this paper it is proposed to extend CVEs through the integration of different collaboration tools in order to allow collaboration to take place in different contexts (e.g. *Web* browsing), in a parallel and coordinated way with the virtual scene.

1. INTRODUCTION

The current World Wide Web has become the major platform for communication and collaboration. The increasing demand for collaboration technologies is an evidence of a common desire to abolish the previous "lonely web" paradigm. One of the main motivations encouraging these technologies is the distributed fashion nowadays companies and research centers work. As a response to that demand, several collaborative systems (also known as Computer-Supported Cooperative Work or CSCW systems) have been proposed.

Collaborative Virtual Environments (CVEs) represent an important category of CSCW systems that provides collaboration facilities through the implementation of a distributed virtual space. The virtual space may be represented from text-based environments to rich 3D shared spaces (also called virtual reality worlds). The use of Virtual Reality (VR) is encouraged due to its great capacity of modeling and interactivity, enabling CVEs to solve lots of CSCW issues like: (i) peripheral awareness (perception of coworkers activity); (ii) representation of real world metaphors (human gestures) and artifacts (3D simulations); (iii) reduced network transmission costs.

However, the majority of the known CVEs consist of

proprietary solutions working independently of other collaboration tools, providing collaboration only inside the proprietary VR world. Accordingly, rich functionalities offered by edition, communication and other collaboration tools are just not provided. The work presented here proposes an extension of CVEs through the integration of third party tools. To accomplish this extension an "Integration Framework" describing how different collaboration tools are to be interfaced to CVEs is defined.

This article is structured as follows. Section 2 presents the related work regarding some existent collaborative systems. Section 3 presents the motivations guiding this integration proposal as well as some integration scenarios. An "Integration Framework" and a system architecture based on it are described in section 4. The fifth section presents some conclusions and directions of future works.

2. RELATED WORK

A number of collaborative systems have been reported in the literature. Most of these systems have the *Web* as underlying architecture. *CoLab* [1], for example, is a multi-user browsing system where users visit pages in a synchronized way. Some *Web*-based applications also provide integrated communication tools, from chat to videoconference (*PageTogether* [2]), and application sharing/whiteboard facilities (*NetDIVE* [3]).

Regarding the CVEs, many efforts have been made to develop 3D based collaborative applications. Systems like *DIVE* [4] and *Open Community* [5] provide means to design complex VR worlds (the core of the collaboration) where people go to interact with each other. VR multi-user environments may be developed through proprietary solutions (*MASSIVE-3* [6], *DIVE*) or through extensions to existent standards and formats (*COSMOS* [7], *Open Community*). Among these standards and formats the most important are: (i) *MPEG-4* [8] (an ISO/IEC standard that uses an object-based approach and a binary language for scene description); (ii) *VRML* [9] (also an ISO/IEC standard scene description language in a text file format); and (iii) *X3D* [10] ("next generation" of *VRML*, still as a "Draft", that can use *XML* to describe the scene).

Among the existent CVEs, only few solutions propose some integration of VR world with external applications. In [11] authors propose the integration of *CRACK!* (it

⁺ Researcher supported by a scholarship from CNPq, Brazil.

^{*} Researcher of the Universidad Veracruzana, México, supported by scholarship 70360 from CONACyT/PROMEP, México.

[®] Research partly supported by the IST project *Lab@Future*.

provides awareness of others on the same *Web* page) and VIVA (a virtual reality application) in order to enable users in the *Web* to be aware of others in the VR worlds. In [4] the virtual environment is interfaced with a text editor/e-mail application allowing users to take notes and post them in a VR mailbox. *Virtual Worlds Platform 5* [12] provides a richer integration where documents of different formats are viewed inside the VR world and VoIP communication may be established. The *MOVE* environment [13] is one of the few solutions proposing an extensible infrastructure to provide collaborative services. However it just supports the integration of new applications developed using its own components.

3. INTEGRATING DIFFERENT COLLABORATION TOOLS WITH VIRTUAL REALITY

As described in the previous section, the integration of different applications with CVEs is not a well-covered topic, even if this integration could provide a number of desirable features. In this section the main motivations for this work and some integration scenarios are explored.

3.1. Motivations

The motivations that guided the development of this work are based on the fact that most existent CVEs provide collaboration among users just into the VR context. However this is a limited way of offering collaborative services.

Let us first start with the *Web*, which is the hugest public information resource. Indeed, it is strongly desirable that any collaboration activity has access to it. The integration of a collaborative *Web* browsing system with a CVE would lead the collaboration activity not to be restricted to the VR context. Thus, users would be able to coordinate and synchronize their actions in both contexts. For example, while visiting a VR museum, users may also browse the exhibition *Web* pages. Following a link to a page describing a specific antiquity takes the user into the VR room where the 3D model of this antiquity is placed.

Another limitation of some CVEs is using the VR interface to show or handle documents (e.g. [13]). As VR was not originally conceived to present specific data like formatted text neither complex numeric spreadsheet, it is a challenge for users to edit/view these documents inside the 3D environment. The integration of external edition tools is then required to enable users to handle these documents properly (instead of trying to edit them inside the virtual scene). Moreover, the integration of application sharing tools would meet current collaborative edition demands.

The last aspect taken in to account in this work deals with the communication needs collaborative work may address. As communication is a key feature for collaboration, it should be inherent in CVEs. Several

CVEs already provide communication means like chats and voice [5,13]. But generally it consists of proprietary solutions and only users connected to the same VR environment can be reached. The integration of external communication tools would allow not only communication among connected users, but also with non-connected ones.

3.2. Integration Scenarios

The first two integration scenarios deal with the integration of CVEs and *Web*:

a) *Bi-directional Synchronization* - In conventional VR documents browsing actions can be executed as a result of user interactions. This is achieved through the definition of anchors on objects composing the VR scene. Once an object is selected, the target URL (pointing to any WWW resource) defined for the respective anchor is then presented. The definition of anchors into VR scenes is actually the only standard feature offering some sort of integration between VR worlds and the *Web*. This integration scenario aims to extend the existent "one-way" integration between VR worlds and *Web*, in order to allow interaction in the other direction. That is, *Web* browsing events could also fire actions into the VR world, changing thus its state (e.g. inserting a new object inside it).

b) *"Web Portal" to VR world* - As the group activity takes place in a VR context, users must have suitable devices to access it ("high-performance" machines and appropriate 3D software). Thus, this will constrain the access of "not VR-able" users. This integration scenario aims to allow users to follow the group activity from outside the VR context. Dynamic *Web* pages representing the current state of the group activity could play the role of mirrors of the VR world. A simple example is a page displaying all the users connected to the VR world and a list containing all the interchanged text messages. Combining the integration scenario described before, this integration could be still enriched. For example, enabling not-connected users to interact with VR world sending messages to be displayed inside it through HTML forms.

The next two integration scenarios aim to extend the functionalities of VR applications making use of other collaboration and communication tools:

c) *Edition tools + Application Sharing* - Inside VR worlds some VR objects may be created to represent real documents (like spreadsheets or formatted text). In conventional VR scenes, the definition of anchors on these objects would allow users to click on them and then to download them or to individually visualize them into a local browser window. In this integration scenario the possibility of executing a collaborative edition tool directly as a result of an interaction inside the VR world is proposed. As there is always a central version of the document (instead of a local version for each user), its VR representation could also reflect the changes made to it.

		Web Integration		Integration of other Applications	
		Bi-directional Synchronization	"Web Portal" to VR world	Edition Tools (+ Application Sharing)	Communication Tools
Training / E-learning	Virtual Classes	Teachers driving lectures both through web and VR browsing.	Allow the access of students who do not have any VR plug-in.	VR objects could be collaboratively edited. Integration with whiteboard application.	-----
	Product Maintenance	Users could execute actions on the VR product as a result of browsing its HTML manual.	-----	-----	-----
Group Work	Product Engineering	-----	-----	Using specific CAD tools to design products, while their VR representations are updated in the VR scene. Similarly to spreadsheet represented by VR charts.	-----
	Group Meetings	Insertion of VR objects representing documents just uploaded through a HTML page.	Redundant information showing meeting activity state to "not-VR" users.	VR whiteboard bound to a real whiteboard tool. (e.g. each 5s the VR one shows the state of this last)	Using VoIP application to contact people not connected to the VR meeting.
E-business	Virtual Products	Customers choose objects from web catalogs and add them into the VR world.	-----	-----	-----
	Virtual Shopping	Interactively visualizing merchandises (both through web browsing and/or VR browsing).	-----	-----	-----

Table 1: Example of applications of the Integration Scenarios

d) *Communication Tools* - The integration of communication tools into VR worlds proposed by this integration scenario aims to facilitate users to communicate more naturally. The main idea is to allow communication tools to be executed as a result of users interactions inside the VR, for example, selecting an users' avatar (a graphic representation of a person) would establish an audio conference with him. Moreover, the integration of communication tools may be related to communication objects. A VR object representing a telephone could actually work as a link to a VoIP application, allowing users from the VR world to call people that are "outside".

In order to illustrate the integration scenarios previously described, it is presented in Table 1 some examples of their use in the context of different collaborative applications.

4. IMPLEMENTING THE INTEGRATION PROPOSAL

4.1. Integration Framework

This integration framework aims to guide the integration among client-server CVE's and different client-server environments. The decision of considering only the client-server paradigm is based on two factors: (i) most of existent CVE's are client-server based; and (ii) the *Web* itself has an intrinsic client-server nature.

As Figure 1 illustrates, to accomplish the desired integration this framework proposes the implementation of an *Integration Module* interfacing with the concerned servers. Its task is basically to monitor the events detected by each server (like user interactions) and execute actions

through these servers. Therefore, this module will define a list relating events to actions. Once an event is notified, its related action message is sent to the correspondent server, who is in charge of executing it.

Inside the VR world, there are three main types of VR events that could be observed during the collaboration activity: selection, creation and changes in the state of objects/avatars. The VR actions to be executed are quite the same. The other events and actions that may be notified or executed will depend on the application of each collaboration server. For example, for a *Collaboration WEB Server* the browsing events are: entering a new URL; following a link in the current page; and submission of a HTML form. Browsing actions basically consist of pushing URLs to force web browsers to navigate.

4.2. Example Architecture

To implement the proposed integration framework, it has been defined a system architecture (see Figure 2) where different collaborative applications are to be integrated. Some requirements for the definition of this architecture were (i) platform independence and (ii) extensibility of systems. As Java is a platform-independent and object-oriented language, it was chosen as base technology in order to meet these requirements.

The *Integration Module* here is a Java application that communicates with each server via *JSDT 2.0* (servers may be in the same machine or distributed). It implements different *event listeners* that are programmed during its initialization (through a configuration file containing an event/action list) to send action messages to the servers.

Concerning the CVE, it was chosen the VNet environment [14], a client/server application implemented

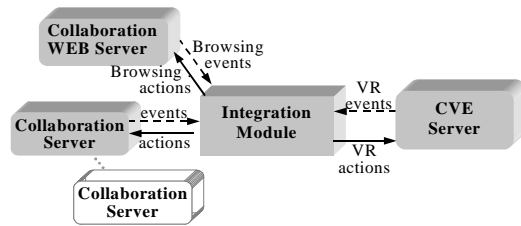


Figure 1: Integration Framework

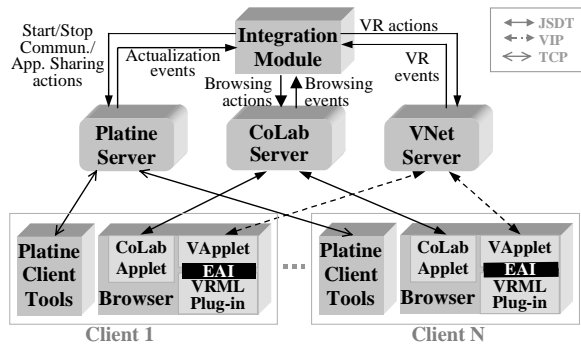


Figure 2: An architecture based on the Integration Framework

in Java that provides the sharing of VRML scenes. The communication between the server and each client is accomplished through the VIP protocol (VRML Interchange Protocol). This is a simple protocol that implements the transmission of VRML fields. At the client side there is an Applet communicating through EAI [15] (EAI defines an interface between a VRML world and an external environment) with a VRML plug-in. This applet is responsible for detecting changes on the VRML objects and communicating them to the server. Once information about these changes arrives, the server broadcasts them to the other applets that update the VR world.

This architecture restricts thus the VR world to be specified through VRML. Actually, this is an acceptable constraint since VRML is a largely used standard, and with the establishment of X3D the migration to this last tends to be direct.

The other two collaborative applications are *CoLab* [1] and *PLATINE* [16]. *CoLab* is a Java-based software environment implementing a collaborative browsing system. The *CoLab Server* is basically composed by a *Web Proxy* (responsible for tracking all browsing activity) and a *Collaboration Engine* (responsible for dynamically defining access rights to web resources based on policy rules). *CoLab Client* is a Java applet through which the server can synchronize visualization of Web pages.

PLATINE is a platform that provides communication tools, like chat, audio and videoconference, and collaboration tools, like shared whiteboard and an application sharing tool. All modules are Java based, except the application sharing server, written in C++. The *Integration Module* controls *PLATINE* tools through start/stop actions. Among these tools, only application

sharing and whiteboard might produce events to indicate eventual changes on shared documents.

5. CONCLUSIONS AND FUTURE WORK

In this paper, it was proposed the extension of collaborative virtual environments through the integration of different collaboration and communication tools. This integration allows the collaboration activity, generally restricted to the VR context, to also take place in different contexts (e.g. web and application sharing) in a parallel and coordinated way. An integration framework and a general architecture based on it were also described.

The ongoing work concentrates on the implementation of the architecture proposed here in order to check its feasibility. A VR collaborative application in the E-learning domain defining a set of VR scenes has also been developed for the purpose of exploring the functionalities provided by the integration proposed in this paper.

6. REFERENCES

- [1] G.J. Hoyos-Rivera, R.L. Gomes, and J.P. Courtiat, "A Flexible Architecture for Collaborative Browsing", 11th IEEE WETICE, MIT, Cambridge, Massachusetts, 2001.
- [2] K. Jiyong *et al.*, "PageTogether: Web-Based Cooperative Work System Using Multi Protocol Interface", Proc. International Conference on Computers in Education, 2001.
- [3] <http://www.netdive.com/indexea.htm>
- [4] E. Frécon and A.A. Nöu, "Building distributed virtual environments to support collaborative work", Proc. ACM Symposium on Virtual Reality Software and Technology, 1998.
- [5] <http://www.opencommunity.com/>
- [6] <http://www.crg.cs.nott.ac.uk/research/systems/MASSIVE-3/>
- [7] V. Darlagiannis and N.D. Georganas, "Virtual Collaboration and Media Sharing using COSMOS", Proc. CSCC 2000, Greece, July, 2000.
- [8] "MPEG-4 Overview", ISO/IEC JTC1/ SC29/WG11/N2725, <http://www.cselt.stet.it/mpeg/standards/mpeg-4/mpeg-4.htm>.
- [9] VRML (ISO/IEC 14772-1), "Virtual Reality Modeling Language", April, 1997.
- [10] <http://www.web3d.org/TaskGroups/x3d/specification-2002february/>
- [11] S. Pekkola, *et al.*, "Collaborative Virtual Environments in the Year of the Dragon", Proc. CVE'00, ACM, Monterey, San Francisco, p.11-18, 2000.
- [12] Blaxxun interactive, "Virtual Worlds Platform 5: product specification", Munich, Germany, 2001.
- [13] García, P. et al "MOVE: Component Groupware Foundations for Collaborative Virtual Environments", CVE'02, ACM, Germany, 2002.
- [14] <http://www.csclub.uwaterloo.ca/u/sfwhite/vnet>
- [15] The VRML External Authoring Interface Committee Draft ISO/IEC 14772-2
- [16] V. Baudin *et al.*, "Supporting distributed experts in e-meetings for synchronous collaboration", SMC'02, IEEE, 2002.