RECOLLVE - REpresenting COLLaboration in Virtual Environments

José Valentim dos Santos Filho
CECOMP/UNIVASF
Juazeiro, Bahia - BRASIL
Email: jose.valentim@univasf.edu.br

Roberta Lima Gomes
DI/UFES
Vitória, ES, BRASIL
Email: rgomes@inf.ufes.br

Aloysio de Castro Pinto Pedroza
GTA/COPPE-Poli/UFRJ
Rio de Janeiro, RJ, BRASIL
Email: aloysio@gta.ufrj.br

Jean-Pierre Courtiat
LAAS/CNRS
Toulouse, França
Email: courtiat@laas.fr

***Abstract*—The world globalization process, the increasing demand of communication and information technologies creates an enormous demand for collaborative applications. Aiming to support communication, information change and collaboration between users distant geographically, many CSCW (Computer Supported Cooperative Work) systems like audio and videoconference, shared editors, shared whiteboard and CVEs (Collaborative Virtual Environments) were developed. Collaborative Virtual Environments represent an important category of CSCW systems that use 3D shared spaces in order to support collaborative activities. We find also many systems that integrate collaborative applications to satisfy the specific needs of users. We propose in this article RECOLLVE, a CVE agent-based which focuses on representing in the virtual scene a very large set of collaboration activities, including their social aspects.**

I. Introduction

Users geographically distant need to change information, to communicate and to cooperate. This context creates an enormous demand for collaborative applications. Currently, many collaborative applications are available for users, such as audioconference and videoconference tools, shared whiteboard, shared text editors, etc. We find also groupwares such as the PLATINE [1] system that groups a set of specific tools, trying to anticipate users needs. However, groupwares like Platine find difficulties to be accepted by work groups, because is practically impossible predict the way how people work together. Consequently, it is practically impossible for developer's predefine real user's collaboration needs and artifacts for work in-group [2]. This fact explains an increasing number of systems to integrate collaborative applications, such as the LEICA system [3], for example. This kind of system provides for users a great flexibility to establish collaborative sessions, allowing users to customize tools according to their needs and preferences.

However, one of the most critical problems on the deployment of cooperative work systems is to provide to user a real perception of presence (awareness) of any user he or she cooperates with. Virtual reality allows to represent the real world by metaphors, making awareness more natural, with the possibilities of interaction very close to those in the real world. In this context, the CVEs play a very important role, because they allow users over many network points to communicate and to interact in a shared 3D environment, often called "virtual environment".

We propose RECOLLVE, a CVE agent-based able to represent collaborative activities. The interaction model proposed allows to representing any collaborative activity. Moreover, the object model proposed includes accessibility rules, which make it possible represent in a virtual scene the access rights policy defined to the virtual collaborative session. We do not have the pretension of solves the access control problems in virtual collaborative sessions. We just want make our CVE able to represent in the virtual scene the users roles, the rights, the rules and the access control police defined in the collaborative session configuration. This aspect is very important because its aim is to transcribe for the virtual scene a natural aspect of real systems: the social protocol mastering the real collaborative work.

This article is structured as follows: Section II presents a state of the art in collaborative virtual environments grouping them by the application type. In Section III, we present our architecture multi-agents; after that, in Section IV, we present the interaction model defined, which allow representing collaborative activities; in Section V a two applications integration experience is related; last section, some perspectives about future works are presented.

II. Related Work

Collaborative virtual environments have been used in many applications type, such as: e-learning, training, war simulations, etc.

EVE[4], *BrickNet*[5], *INVITE project* [6], *DIVE* [7], *VREng* [8] and *VNet* [9] are examples of CVEs where users can share a 3D environment and interact with others users and objects.

DIVE is a virtual reality desktop system where users can dynamically program behaviors to objects and avatars. However, avatars have no knowledge about objects inside virtual scene. DIVE provides a very restricted integration with others collaborative applications.

VREng (*Virtual Reality Engine*) [8] is a virtual reality distributed system that allow to users interact and move in 3D world. *VREng* provides also a closed integration of its virtual world with others collaborative tools such as a whiteboard. However, user's actions are not represented in the virtual scene by avatars actions.

Soares et al. [10], propose a 2D-3D hybrid interface. User's actions on 2D application shared interface in a populated virtual world are represented in the 3D space.

NETICE [11] allow sharing a whiteboard in a 3D interface and providing avatars with a set of facial and body expressions. But, these avatars actions are realized from an actions menu, independent of actions realized on the *whiteboard*.

The main idea in *VEPersonal* [12] is to construct virtual environments user knowledge level adapted. A same virtual scene presents different details for users with different knowledge level. However, *VEPersonal* does not represent collaborative applications.

Second Life [13] and *There* [14] provide their users with very detailed and advanced avatars that are often also extremely customizable. Users choose between many virtual worlds to navigate and interact with others users by a chat system. However, the avatar-object interaction possibilities are limited to a same and reduced set of predefined actions.

INVITE system is a platform for tele-learning that provides a shared whiteboard, support multi-modal interaction between users by text, video and audio communication. Users have different roles. *INVITE* provides an integration of virtual environment with others collaborative tools in a closed system. The *INVITE* system is the more closed of RECOLLVE, but it does not considers access rights by users.

A. Comparative between CVEs

	Integration	Roles	Rights	Awareness
INVITE	•	•		•
DIVE	•			
VREng	•			
Netice	•			
EVE		•		•
Soares et al	•			

From comparison between CVEs early referred, we conclude that there are virtual environments able to represent specific collaborative applications, as a white board, for example, but no one provide to users mechanisms to represent any kind of collaborative activity, including their social aspects.

We also found systems agent-based like InViWo [15], Agile [16] and PAR [17]. In all of them, we identify very interesting characteristics. However, no one of them focuses on represent collaborative activities. In this paper, we present RECOLLVE - a collaborative virtual environment agent-based that provides mechanisms to users represented colaborative applications.

In next Section, we present the architecture agent-based of RECOLLVE that model the behavior of avatars and objects-agents and the relationships between them.

III. RECOLLVE - The Architecture

RECOLLVE is a CVE agent-based focused on represents collaborative applications in virtual environments. The Figure 1 shows the architecture multi-agents of RECOLLVE. In

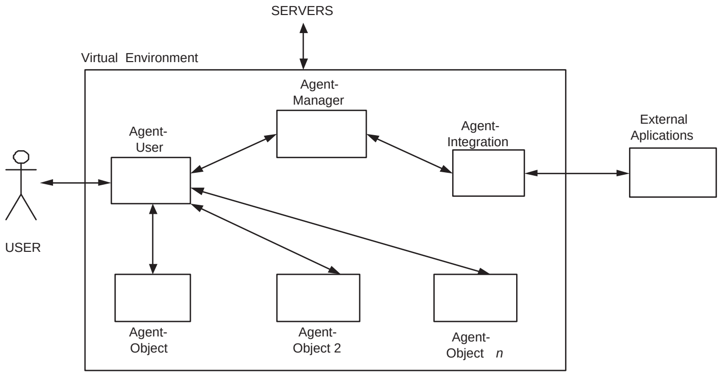

Fig. 1. Multi-agent Architecture of RECOLLVE

RECOLLVE, all entities are represented by agents which to communicate and to interact between themselves.

We have defined four types of agents:

- agent-user: users are represented in the virtual scene by embodied *Agents-Users*, here we referred it by avatar. The *Agent-User* model is very simple: when an avatar is coming near an object or other avatar, it sends one message to that agent informing its profile. When an avatar receives some message from others agents, it presents the content message to user.
- agent-object: objects are represented in the virtual scene by embodied *Agents-Objects*. Objects define what actions an avatar can realize on it, based on early defined rules. The Figure 2 shows the interaction between an avatar and object agent. The avatar sends a message to object which it wants handle informing its profile. The object evaluates based on established rules what actions that avatar can realize on it. It sends back to avatar a message with the available actions set. Then, an actions menu is presented to user and he or she make a choice. In this way, it is possible transcribe for the virtual scene the access rights control of users, according to his profile. When an object is handled by an avatar, it may be moved by the avatar.

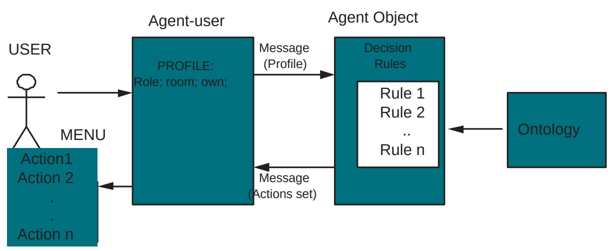

Fig. 2. Agent-user and Agent-object interaction

- agent-manager: it reacts to events inside virtual scene and realize actions based on predefined rules. The 3D scene is divided in rooms, where to each room we assigned one agent manager.
- agent-integration: serve request from external applications to realize actions inside virtual scene and vice-versa.

We chose handling objects and avatars in a same way because communication facilities between agents provided by standard languages.

IV. Interaction Model to support Awareness

Awareness represents the common conscience of a work group that knows the general context of work. This conscience group corresponds to understanding of a user have about: i) which whom he works with; ii) what each user does; how do users actions interacts [18].

We analyzed a list not exhaustive of collaborative applications, where we can identify many actions, which are commons in almost all of them: be aware of who is doing what; the transfer permission to realize an action; the queue to protagonist the action, etc:

- audioconference: the use of audioconference system for many users as the unique communication way presents difficulties to aware the speakers, to aware who has the permission to talk, to aware who waits to talk, etc.
- shared editor: allows a users group geographically distant to share an editor. The main difficulty in this kind of system is to manage concurrent tasks. Aware whom has the permission to edit.
- collaborative browsing: allow to a users group browsing together in the Web. The master role drives the browsing. The master role is dynamic, and then a user can pass this role to another user. Here again, the main difficulty is aware the master, aware who follow the master, etc.

Therefore, the challenge is to develop an interaction model, which allows representing any collaboration activity in a virtual scene.

Kallman and *Tallman* [19] proposed a general interaction model between avatars and objects in the virtual world. The main idea is that all needed information for an avatar to interact with an object be included in the object itself. This proposes is focused into an avatar x object interaction.

Jorissen et al [20] extended the Kallman and Tallmann model to any object which interacts with any other, making not any difference among objects, avatars or autonomous agents. This propose is focused on object x object interaction.

From observation of collaborative applications such as shared text editors, audioconference tools, collaborative browsers, we evidence that Kallmann and Jorissen models are not able to provide some kinds of interaction, presents in many kinds of collaborative tools. For example, an avatar gives a 3D object representing a microphone to another avatar symbolizing transmission the permission to talk.

This interaction example goes beyond of capacity Avatar x Object and Object x Object interaction models. The interaction model proposed in this work include characteristics from the two other models described above and extend them to provide interactions between N entities. Each action is propagated by all entity chain long. Some examples of interactions types described by our model:

- avatar/agent x object x avatar/agent: an avatar gives a 3D artifact to another avatar illustrate this kind of interaction; scenario example: an avatar gives a 3D object to another avatar, representing the toking pass to another user;
- avatar/agent x object x object: actions which avatars using a 3D object to interact with another 3D object; scenario example: an avatar uses a 3D object representing a pen to "write" in another 3D object representing a text editor.

The classic solutions define mechanisms only for some interactions types. In the other hand, our approach is based on an interaction abstract model.

The aim here is to map real actions into virtual actions, or to map user's actions in avatars actions. Avatars make awareness between users stronger and collaboration activities become clearer when expressed by avatars actions. Because the interaction model defined, our avatars realize suitable actions on objects, respecting their functionality.

An actions library is available:

- *Navigation actions*: are used in the avatars' movement in the virtual scene, like to walk, to run and to jump;
- *General actions*: this kind of action is executed on the virtual objects without consider its functionality, e.g. to get it, to release it, to give it, etc;
- *Suitable actions*: this kind of action is executed over virtual objects considering its functionality, e.g. to sit down, to read, to call, etc;
- *Facial Expressions*: are used to represent users emotions;
- *Gestural*: are used to indicate desire to realize a specific activity.

For your turn, real applications are mapped in 3D-objects and each 3D-object belongs to a class according to its function in the virtual scene. For example, objects used by avatars to realize an action are called Token. All Objects in this class have the same set of actions, like: to take, to release, to give.

A. Implementation Issues

The first version of RECOLLVE prototype has client/server architecture as shown in Figure 3. RECOLLVE uses the *VIP - VRML Interchange Protocol* [9] to assure the communication between clients and the server; VIP has been extended with rights manager and actions messages. The interface allows to user connected to the virtual world, represented by an avatar, to navigate through it and interact with others users and objects.

Our client is made up of a Java desktop GUI and a Xj3D-based browser [1]. The GUI is used to connect users in the virtual scene session. Moreover, from GUI a user can put or remove objects if his role allows. The Xj3D is a plug-in used for user's navigation and access to a 3D world.

The Server is responsible for handle connection request, for the initialization of each new client in the multi-user virtual world. It is also responsible for transmits the current state of the 3D scene to the newly added client, as well as for sending update messages regarding the avatar position and orientation in the multi-user virtual world.

[1] http://www.xj3d.org

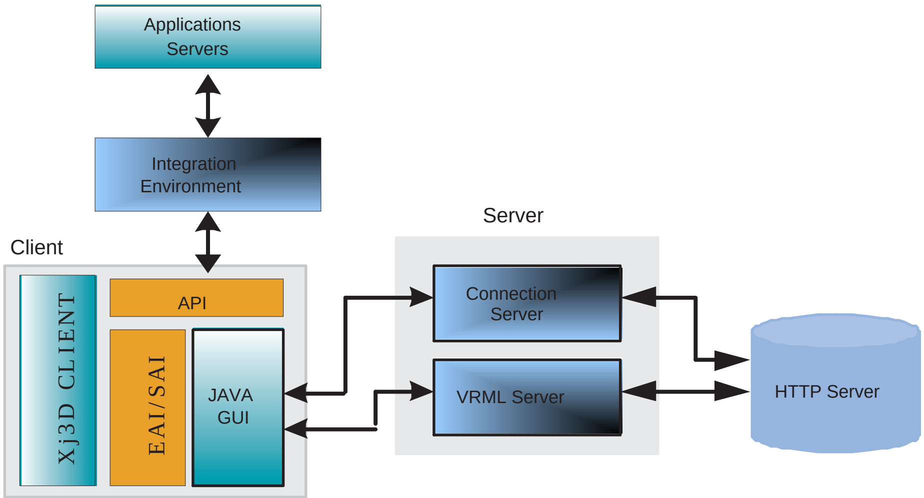

Fig. 3. RECOLLVE Implementation Architecture

1) API: In [21], *Gomes et al.* argued that miss of an API (*Application Program Interface*) in the collaborative tools is one of the main problems to make a suitable integration between collaboration tools. In a CVE, the complexity to implement this kind of system force the developers to choose proprietary solutions, which makes it more difficult its integration to others tools separately developed. Thus, the majority of CVEs works independently of others collaborative applications and the collaboration happen only inside the virtual scene.

The API allow integrate RECOLLVE to others collaborative applications. In this way, extern events have consequences in the virtual scene. In the same way, events in the virtual scene have consequences in external applications. For example, the avatar selection of another user could start automatically an audiconference session, and a 3D object representing a microphone would be put into the virtual scene; or a 3D object representing a phone could play the role of a link for a VoIP application, like the Skype systems. Thus, a Skype session could be represented in the virtual scene.

The integration system has the same capabilities to realize actions as a normal user. Some actions available by the API are described below:

- add a 3D object in the virtual scene as a consequence of an external action;
- make an avatar take a 3D object;
- make an avatar "gives" a 3D object to another avatar;
- remove a 3D object from the virtual scene as a consequence of an external action.

B. Sharing VRML Worlds

From the technical viewpoint, VRML2.0[22] is the standard technology for the creation of VEs. It has been supported by various editors and 3D authoring tools like VRMLPad[2] and X3DEdit[3]. However, VRML does not provide support for multi-user virtual worlds. The VRML model was defined to satisfy a unique scene graph. Nevertheless, in our case, each user has your own not shared scene graph, it running in VRML plug-ins over different machines.

A VRML scene is organized as a direct acyclic scene graph where each node is able to produce and receive events. A *TouchSensor* node put out a *SFTime* event always that a user click on object. This event retrieves the exact instant of the click. Another node receives this event as an *eventIn* and starts an action.

The ROUTE mechanism allows transmiting an event from a node as an *eventOut* to another node as an *eventIn*. A ROUTE is defined like that:

```
ROUTE node1.eventOut1 TO node2.eventOut2
```

In order to point out the incoherence's of events propagation in multi-user VRML world, we use as an example a multi-user VRML world, which contains two avatars representing two users and a chair, upon that avatars could sit on. When a user clicks on the chair, *eventOuts* will be produced on the node *TouchSensor*, and routed to *eventIns* of avatars. Thus, a correspondent animation will begin. The structure desired is shown in Figure 4

However, the real situation is described in Figure 5, because events produced by chair cannot make difference between two avatars A and B, and the two avatars or neither one will sit on the chair.

Thus, we conclude that route mechanism of VRML cannot be used for the multi-user case. There is a need for a multi-user extension behind the VRML. Many approaches like LW[23], VSPLUS[24], Spin-3D[25], and EVE[4] were proposed with the aim of become a standard. In RECOLLVE, we implemented our own approach to VRML data sharing.

Events generated by entities in the virtual scene are retrieved by EAI/SAI [26][27] and forward to a routing table, as

[2]http://www.parallelgraphics.com/products/vrmlpad/
[3]http://www.web3d.org/x3d/content/README.X3D-Edit.html

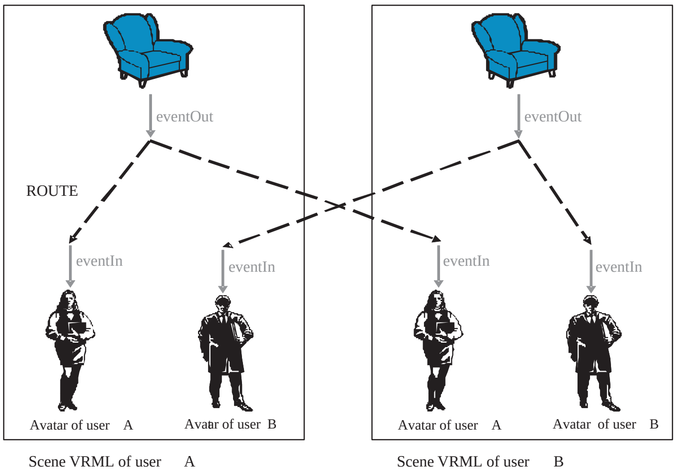

Fig. 4. Events Synchronization - Desired Situation

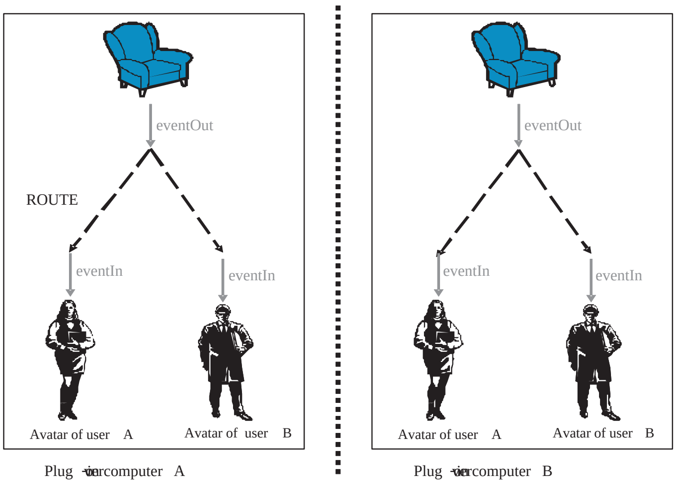

Fig. 5. Events Synchronization - Real Situation

shown in Figure 6. This routing table allows to define what *eventIns* of what entities will receive the produced event in the virtual scene. From VRML perspective, this procedure is transparent, but from perspective of system, the routing table store information has to make difference clients by their entities. The aim is add a property of object-oriented system to a language that does not have it.

The Figure 7 shows the case of a user click over a chair. The produced event by chair's click allows to routing table compose an identifier pair, which is composed by an ID, unique identifier to each entity, and an index that identify an *EventOut* for that entity. After, the table will be initialized with all possibles couples. The value of initial event will be send to correct entity because these couples. In this case, only the avatar of user A will receive a command to start animation "sit on chair"

Our approach creates an interface for VRML objects. Objects are composed by a 3D description and an interface,

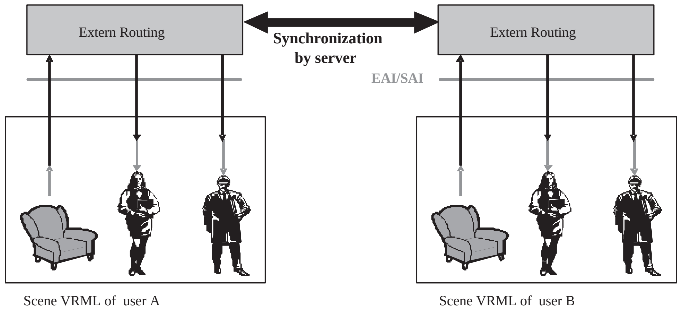

Fig. 6. Events Synchronization by Server

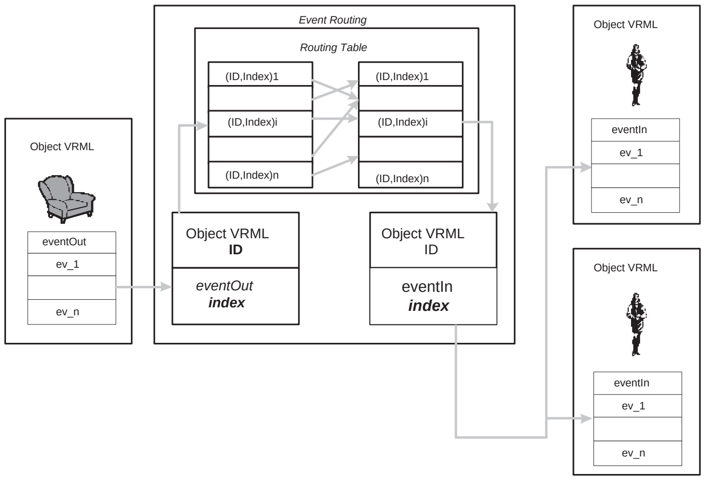

Fig. 7. Routing Table

as shown in Figure 8. The object interface defined by the system administrator has interactions information's, behavior informations, attributes and accessibility rules. The interaction information's are the description of what actions are available for the user and his avatar.

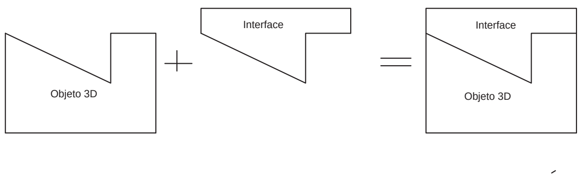

Fig. 8. Object composition

Our avatar model follows the *H-Anim* standard (Humanoid Animation Working Group)[4]. *H-Anim* is a standard to describe animated humanoids. A H-Anim body is a hierarchical set of articulation nodes to which others articulation nodes can be associated. Thus, to displace *H-Anim* body's part is need to set new localization values to the articulation nodes concerned in the displacement. To specify an action following *H-Anim* standard is a not intuitive and very complicated task because the distance between *H-Anim*, an animation language and the

[4]http://www.h-anim.org

natural language.

V. Case Study - Supporting Awareness in a Collaborative Browse Session

This section describes the general context of work presented in this article. Figure 9 shows an integration scenario where two collaborative applications - our CVE prototype and COLAB [28] - are integrated by LEICA [3].

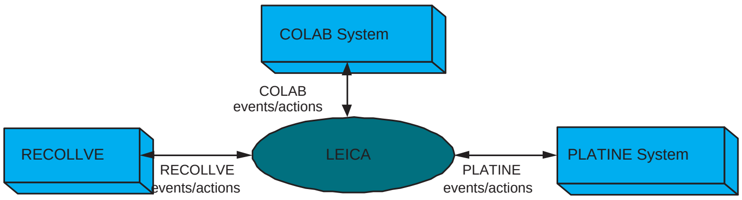

Fig. 9. A possible integration scenario

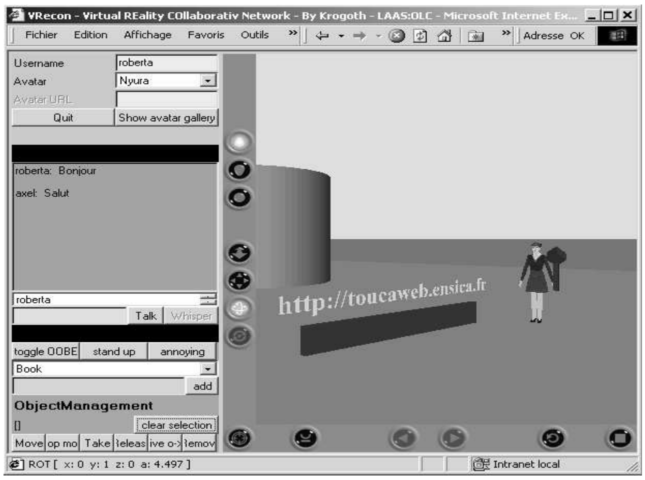

Fig. 10. Representation of COLAB in CVE

Fig. 11. URL loaded in the user's browser

LEICA is a system of integration rule-based where the applications (CVE and COLAB in this example) interact by messages events notification.

COLAB is a collaborative browsing system in the Web. COLAB users can create and destroy all the synchronization relationships of a dynamic and distributed way. A COLAB client is an applet JAVA which synchronize and presents the Web pages. In this way, COLAB allows to a set of users: browsing together in the Web (a user drives the browsing) or the users browsing in the Web theirselves, but they have awareness of the others browsing users.

Figure 10 shows a experience realized with the first version of RECOLLVE, where a COLAB browsing session is represented in virtual scene. The first step is to choice states will should be represented in RECOLLVE and the real world metaphors which will be associated with these representations. We choice two main aspects of COLAB:

- Browsing Synchronism - when browsing is synchronous, a user drive it and the others users follow him. This state is represented for a entity, represented by the URL indicator. When the browsing is asynchronous, this entity disappear.
- *Floor* - this expression indicate a unique attribute associated to a user, which allow him to drive a synchronous browsing. This attribute is represented by a 3D object (a token).

When COLAB alternate between synchronous and asynchronous mode, the token appear and disappear. When the URL is modified, its text indicator in the virtual scene is also modified. Figure 11 shows an interface COLAB-Web Page, where the text indicator is shown in the virtual scene, in Figure 10. In COLAB, when a user pass the floor to another user, the token will be passed in the virtual scene too. Finally, if a user pass the token to another user in the virtual scene, the floor will be passed to another user in COLAB too.

VI. Perspectives for Future Work

In this article, we proposed RECOLLVE - a collaborative virtual environment agent-based that provide mechanisms to users represented any kinds of colaborative applications. This is possible thanks to event and interaction model defined. Moreover, we defined an API to collaborate beyond of virtual scene. We realized also a first integration experience to representate a collaborative navigation session inside the virtual scene. For future works, we intend analyze and evaluate the usability of this virtual interface. Currently, the centralized architecture of RECOLLVE is a bottleneck to large simulations. Then, it is needed to define and evaluates with simulation help another architecture.

References

[1] V. Baudin, "Supporting distributed experts in e-meetings for synchronous collaboration," in *Proc. IEEE International Conference on Systems, Man and Cybernetics, SMC'02, Tunisie*, 2002.

[2] J. Hummes and B. Merialdo, "Design of extensible component-based groupware," in *Computer Supported Cooperative Work: The Journal of Collaborative Computing*. Kluwer Academic Publishers, 2000, pp. 53–74.

[3] R. L. Gomes, G. de Jess Hoyos Rivera, and J.-P. Courtiat, “LEICA: Loosely-Coupled Integration of CSCW Systems,” in *5th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS 2005) LNCS Springer-Verlag, Athens, Greece*, Jun. 2005.
[4] C. Bouras and T. Tsiatsos, “Distributed virtual reality: bulding a multi-user layer for the eve Platform,” in *Journal of Network and Computer Applications*, vol. 27. Elsevier, 2004, pp. 91–111.
[5] G. Singh, L. Serra, W. Png, A. Wong, and H. Ng, “Bricknet: Sharing objects behavior on the net,” in *Proceedings of the Virtual Reality Annual International Symposium (VRAIS(95)*. IEEE, 1995.
[6] I. Project, “Web site: http://invite.fh-joanneum.at/, may 2008.”
[7] E. Frecon and M. Stenius, “DIVE: A scalable network architecture for distributed virtual environments,” in *Distributed Systems Engineering Journal (special issue on Distributed Virtual Environments)*, vol. 5, 1998, pp. 91–100.
[8] P. Dax, “Virtual Reality Engine - web site: http://vreng.enst.fr/html/index.html.”
[9] VNET, “Web site: http://www.csclub.uwaterloo.ca/u/sfwhite/vnet.”
[10] P. Horain, J. M. Soares, P. K. Rai, and A. Bideau, “Virtually enhancing the perception of user actions,” in *15th International Conference on Artificial Reality and Telexistence*, 2005.
[11] W. H. Leung and T. Chen, “A multi-user 3-d virtual environment with interactive collaboration and shared whiteboard technologies,” *Journal of Multimedia Tools and Applications, Special Issue on Distributed Multimedia Systems for Virtual Society*, vol. 20, no. 1, pp. 7–23, 2003.
[12] M. S. Aquino, F. F. de Souza, and A. C. Frery, “Vepersonal: an infrastructure of virtual reality components to generate web adaptive environments,” in *WebMedia ’05: Proceedings of the 11th Brazilian Symposium on Multimedia and the web*. New York, NY, USA: ACM Press, 2005, pp. 1–8.
[13] “Second life: Your World, Your Imagination, World Wide Web, http://secondlife.com, may 2008,” December.
[14] “Welcome to There!!, World Wide Web, http://www.there.com, may 2008,” December.
[15] N. Richard, “InViWo agents: Write once, display everywhere,” in *Proceedings of the Web3D Conference*, 2003.
[16] Y. Zhang, L. Guo, and N. D. Georganas, “AGILE: An architecture for agent-based collaborative and interactive virtual environments,” in *Proc. Workshop on Application Virtual Reality Technologies for Future Telecommunication System, IEEE Globecom’2000 Conference*, 2000.
[17] N. I. Badler, R. Bindiganavale, J. Allbeck, W. Schuler, L. Zhao, and M. Palmer, “Parameterized action representation for virtual human agents,” *Embodied conversational agents*, pp. 256–284, 2000.
[18] P. Dourish and V. Bellotti, “Awareness and coordination in shared workspaces,” in *ACM Conference on Computer Supported Cooperative Work (CSCW92)*. ACM Press, 1992, pp. 107–114.
[19] M. Kallmann and D. Thalmann, “Direct 3d interaction with smart objects,” in *VRST ’99: Proceedings of the ACM symposium on Virtual reality software and technology*. New York, NY, USA: ACM Press, 1999, pp. 124–130.
[20] P. Jorissen and M. Wijnants, “Dynamic interactions in physically realistic collaborative virtual environments,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 6, pp. 649–660, 2005, member-Wim Lamotte.
[21] R. L. Gomes, “Leica: Un environnement faiblement coupl pour lintgration dapplication collaboratives,” Ph.D. dissertation, Universit Paul Sabatier - Toulouse - France, 2006.
[22] “VRML (ISO/IEC 14772-1), virtual reality modeling language - 1997.”
[23] W. S. Web3D Consortium, “Living Worlds Working Groups, 1997.”
[24] Y. Araki, “VSPLUS: A High-level Multi-user Extension Library For Interactive VRML Worlds,” in *Proceedings of the VRML’98*, 1998.
[25] C. g. Stephane louis Dit Picard, Samuel Degrande and C. Chaillou, “VRMLData Sharing in the Spin-3D CVE,” in *WEB3D 2002 SYMPOSIUM : Proceedings 7th International Conference on 3D WEB SIGGRAPH, Tempe, Arizona, USA*. ACM Press, Jun. 2002, pp. 165–172.
[26] “VRML EAI (ISO/IEC FDIS 14772-2), the vrml external authoring interface committee final draft - 2002.”
[27] “X3D SAI (ISO/IEC 19775-2:2004), the x3d scene access interface - 2004.”
[28] G. de Jess Hoyos Rivera, “Colab: Conception et mise en oeuvre d’un outil pour la navigation coopratives sur le web,” Ph.D. dissertation, Universit Paul Sabatier - Toulouse - France, 2005.