The Agent-based Architecture of RECOLLVE

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 for communication and information technologies create a demand for collaborative applications. Aiming at supporting communication, information exchange and collaboration among users geographically distants, 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 specific needs of users. We present in this article the agent-based architecture of RECOLLVE, a CVE which focuses on representing a very large set of collaboration activities in a virtual scene, including their social aspects.

1 Introduction

Users geographically distant need to exchange information, to communicate and to cooperate. This context creates a 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 it is practically impossible to predict the way how people work together. Consequently, it is practically impossible for developers to predefine real users collaboration needs and artifacts to work in group [2]. This fact explains the increasing number of systems to integrate collaborative applications, such as the LEICA system [3], for example. This kind

978-0-7695-3500-5/08 $25.00 © 2008 IEEE
DOI 10.1109/SBSC.2008.32

of system provides for users a great flexibility to establish collaborative sessions, allowing them to customize tools according to their needs and preferences.

However, one of the most critical problem on the deployment of cooperative work systems is to provide a user with 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 present RECOLLVE, an agent-based CVE able to represent collaborative activities. The interaction model proposed allows to represent any collaborative activity. Moreover, the proposed object model includes accessibility rules, which make it possible to represent in a virtual scene the access rights policy defined to the virtual collaborative session. We do not have the intention of solving access control problems in virtual collaborative sessions. We just want to make our CVE able to represent in a 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 to the virtual scene a natural aspect of real systems: the social protocol mastering the real collaborative work.

This article is structured as follows: Section 2 presents a state of the art in collaborative virtual environments grouping them by the application type. In Section 3, we present our multi-agent architecture. After that, in Section 4, we present the defined interaction model, which allow to represent collaborative activities. Section 5 presents a two applications integration experience. Some perspectives about future works are presented in the last section.

2 Related Work

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

EVE[4], *INVITE project* [5], *DIVE* [6] and *VREng* [7] and *VNet* [8] 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 a virtual scene. DIVE provides a very restricted integration with other collaborative applications.

VREng (*Virtual Reality Engine*) [7] is a virtual reality distributed system that allow users to 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, users actions are not represented in the virtual scene by avatars actions.

Soares et al. [9] propose a 2D-3D hybrid interface. Users actions on 2D application shared interface in a populated virtual world are represented in a 3D space.

NETICE [10] allows to share a whiteboard in a 3D interface and provides avatars with a set of facial and body expressions. But, these avatars actions are made according to an actions menu, independent of actions made on the *whiteboard*.

Second Life [1] and *There* [2] provide their users with very detailed and advanced avatars

[1] http://www.secondlife.com
[2] http://www.there.com

that are often also extremely customizable. Users choose among many virtual worlds to navigate and interact with other users by a chat system. However, the avatar-object interaction possibilities are limited to a reduced set of predefined actions.

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

2.1 Comparative among CVEs

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

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

The most of visited CVE presents a closed integration with collaborative tools, but no one provide an open integration API, allowing users to use their prefered tools. As regards to the access control rights, RECOLLVE aims at to transcribe the virtual scene and to represent the access constraints of each role defined in a collaborative session.

We also found agent-based systems like InViWo [12], Agile [13] and PAR [14]. All of them present very interesting characteristics, however, no one focuses on representing collaborative activities.

In this paper, we present RECOLLVE - a collaborative virtual environment agent-based that provides mechanisms to represent colaborative applications.

Next Section presents an agent-based architecture of RECOLLVE that models the behavior of avatars and objects-agents and the relationships between them.

3 The Architecture

RECOLLVE is an agent-based CVE focused on representing collaborative applications in virtual environments. Figure 1 shows the multi-agent architecture of RECOLLVE. In RECOLLVE, all entities are represented by agents which communicate and interact among themselves.

We have defined four types of agents:

- user-agent: users are represented in the virtual scene by embodied *User-Agents*, referred as avatar. The *User-Agent* model is very simple: when an avatar is coming near to an object or to another avatar, it sends a message to that agent informing its profile. When an avatar receives some message from other agent, it presents the message contents to the user;

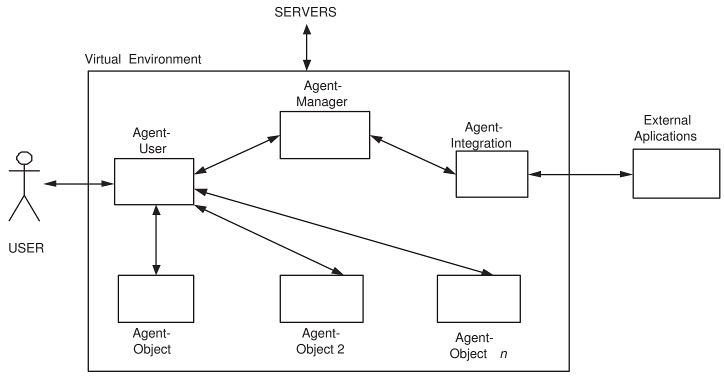

Figure 1. Multi-agent Architecture of RECOLLVE

- object-agent: objects are represented into the virtual scene by embodied *Object-Agent*. Objects define what actions an avatar can realize on it, based on early defined rules. Figure 2 shows the interaction between an avatar and an object agent. The avatar sends a message to the object which it wants to handle informing its profile. The object, based on established rules, evaluates what actions that avatar can realize on it. It sends back a message to the avatar with the available actions set. Then, an actions menu is presented to the user and he or she can make a choice. In this way, it is possible to transcribe to the virtual scene the access control rights of users, according to his profile.

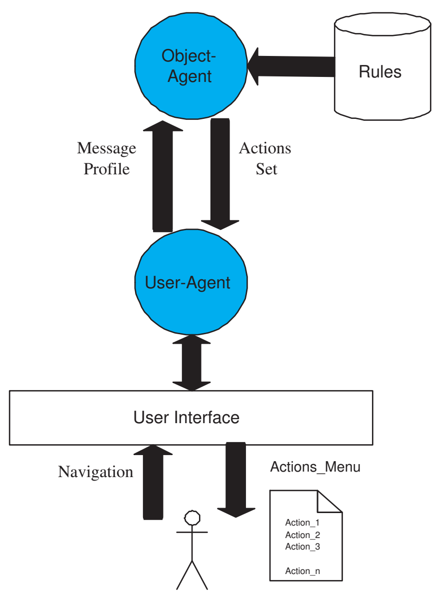

Figure 2. Agent-user and Agent-object interaction

When an object is handled by an avatar, it may be moved by the avatar. In our agent-model, each agent has a mailbox where the message from users agents are posted and the agent is notified. The Object-agent picks-up the message from a message queue to process it. Thus, two or more Users-agent cannot handle the same object at the same time;

- manager-agent: the Manager-agent coordinate actions of other agents; it sniffs actions from users and from theirs agents and it also keeps the virtual scene synchronized among all users. Moreover, it makes actions due to users actions. For example, the manager-agent can exclude an avatar, excluding the user from a virtual session, if it does not respect any rule into the virtual world; or it could order the integration-agent to mute the microphone of a misbehave user.

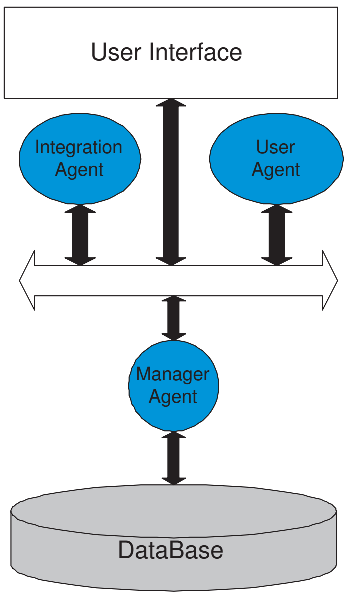

Figure 3. Manager-Agent

- integration-agent: serves requests from external applications to realize actions inside a virtual scene and vice-versa. When it receives one real event notification message from *Integration-Environment*, it asks the Manager-Agent to make a specific action, for example: to put an object into the virtual scene; or when it receives one virtual event notification message from the Manager-Agent, the Integration-Agent asks the Integration-Environment to make a specific action. Figure 4 shows the interactions of the Integration-Agent with the Agent-Manager and Integration-Environment.

Objects and avatars are handled in the same way due to the communication facilities between agents provided by standard languages.

We use the JADE [15] plataform to model and simulate our multi-agent architecture. Each running instance of the JADE runtime environment is called a Container as it can contain several agents. The messages exchanged by JADE agents have a format specified by the ACL language defined by the FIPA - Foundation for Intelligent Physical Agents - [16] an international standard for agents interoperability.

4 Interaction Model to support Awareness

Awareness represents the common conscience of a work group that knows the general context of the work. This conscience group corresponds to the understanding a user has about: i) which whom he works with; ii) what each user does; how do users interacts [17].

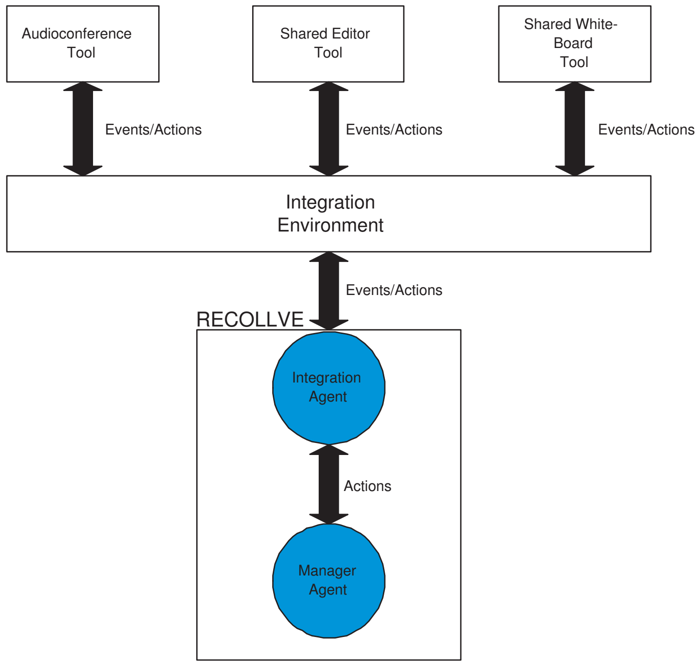

Figure 4. Integration-Agent

We analyzed a non exhaustive list of collaborative applications, where we identified many actions, which are common to almost all of them: be aware of who is doing what; the transfer of permission to realize an action; the queue to realize the action, etc:

- audioconference: the use of an 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* [18] proposed a general interaction model between avatars and objects in a virtual world. The main idea is that all needed information for an avatar to interact with an object should be included into the object itself. This propose is focused into an avatar x object interaction.

Jorissen et al [19] extended the Kallman and Tallmann model to any object which interacts with any other, making no difference among objects, avatars or autonomous agents. This propose is focused on object x object interaction.

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

This interaction example goes beyond the capacity of the Avatar x Object and Object x Object interaction models. The interaction model proposed in this work includes characteristics from the two models described above and extends them to provide interactions among N entities. Some examples of interactions types described by our model:

- avatar/agent x object x avatar/agent: an avatar giving a 3D artifact to another avatar illustrates this kind of interaction; scenario example: an avatar giving a 3D object to another avatar, representing a token passing to another user;
- avatar/agent x object x object: actions where avatars using a 3D object interact with another 3D object; scenario example: an avatar uses a 3D object representing a pen to "write" into another 3D object representing a text editor.

The classic solutions define mechanisms only for some interaction types. On 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 users actions into avatars actions. Avatars make awareness between users stronger and collaborative activities become clearer when expressed by avatars actions. Due to the defined interaction model, our avatars realize suitable actions on objects, respecting their functionality.

An actions library is available:

- *Navigation actions*: are used to move avatars in a virtual scene, making avatars to walk, to run and to jump, for example;
- *General actions*: this kind of action is executed on 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 specific activities.

Real applications are mapped into 3D-objects and each 3D-object belongs to a class according to its function in a virtual scene.

4.1 Implementation Issues

The first version of RECOLLVE prototype has a client/server architecture as shown in Figure 5. RECOLLVE uses the *VIP - VRML Interchange Protocol* [8] to assure the communication between clients and a server; VIP has been extended with rights and actions messages. The interface allows a user connected to the virtual world, represented by an avatar, to navigate through it and interact with other users and objects.

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

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

The Server is responsible for handling connection requests and for the initialization of each new client into the multi-user virtual world. It is also responsible for the transmition of the current state of the 3D scene to the newly added client, as well as for sending update messages regarding to the avatar position and to the orientation in the multi-user virtual world. In RECOLLVE [20] is described how VRML Multi-user Virtual Scene is synchronized and shared.

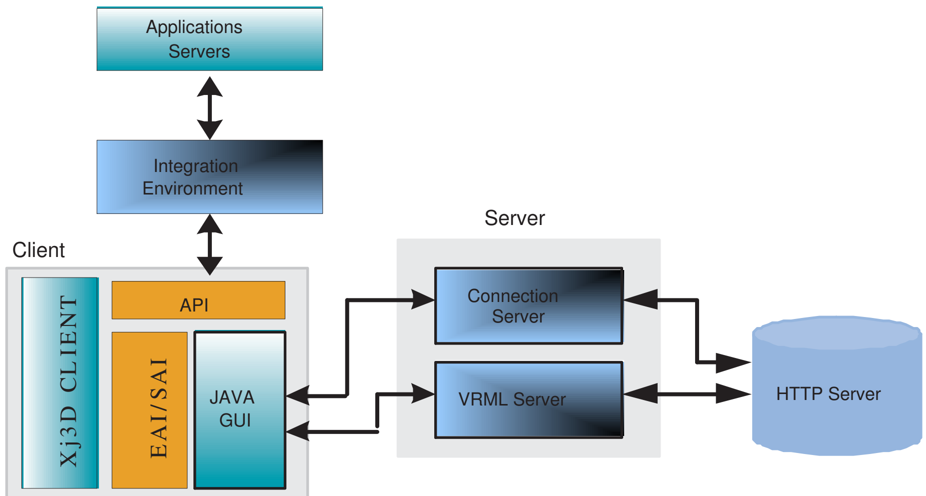

Figure 5. RECOLLVE Implementation Architecture

4.1.1 API

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

The API allow to integrate RECOLLVE to other 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 system. 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 offered by the API are described below:

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

5 Case Study - Supporting Awareness in a Collaborative Browsing Session

This section describes the general context of the work presented in this article. Our Agent-based Architecture is mapped into a virtual scene. The JADE Container which is the agents environment is represented by a 3D Scene. Users and Object-Agents are represented respectivelly by Avatars and 3D objects.

LEICA is a system of integration rule-based where the applications (CVE and COLAB in this example) interact by message event notifications.

Figure 6 shows an integration scenario where two collaborative applications - our CVE prototype and COLAB [22] - are integrated by LEICA [3].

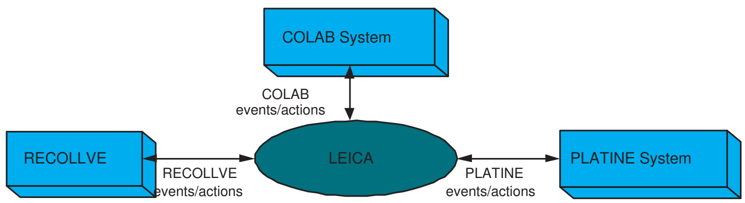

Figure 6. A possible integration scenario

COLAB is a collaborative browsing system in the Web. COLAB users can create and destroy all synchronization relationships on a dynamic and distributed way. A COLAB client is a JAVA applet which synchronizes and presents Web pages. In this way, COLAB allows a set of users: to browse together into the Web (a user drives the browsing) or users to separately browse into the Web, but having awareness of the other browsing users.

Figure 7 shows an experience realized with the first version of RECOLLVE, where a COLAB browsing session is represented into the virtual scene. The first step is to choose states which should be represented in RECOLLVE and the real world metaphors which will be associated with these representations. We choose two main aspects of COLAB:

- Browsing Synchronism - when browsing is synchronous, a user drives it and the other users follow him. This state is represented as an entity, represented by the URL indicator. When the browsing is asynchronous, this entity disappears.
- *Floor* - this expression indicates 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). Accessibility rules are established for the token, defining the user profiles which have access to it.

When COLAB alternates between synchronous and asynchronous mode, LEICA, playing the role of Integration-agent, makes a token to appear and to disappear. When the URL is modified, its text indicator in the virtual scene is also modified. Figure 7 shows an interface COLAB-Web Page, where the text indicator is shown in the virtual scene. In COLAB, when a user pass the floor to another user, a token will be passed in the virtual scene too. Finally, if a user pass a token to another user in the virtual scene, the floor will be passed to another user in COLAB too.

Another case study, under development at RECOLLVE, is to represent a virtual class in the context of *E-leaning*. Teacher and students talk through an audioconference tool,

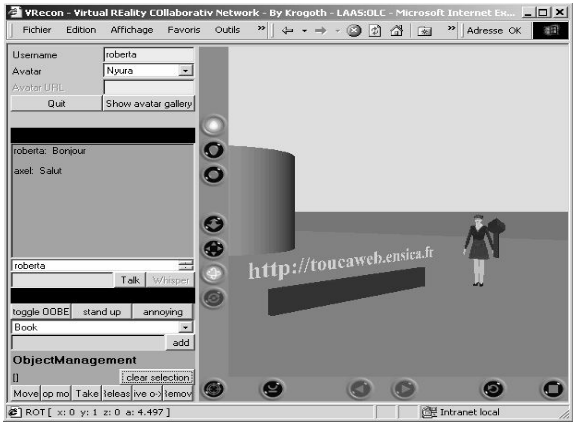

Figure 7. Representation of COLAB in CVE

Figure 8. URL loaded in the user's browser

like the Skype system. However, the main problem with audioconference systems is to coordinate the communication when there are many users participating: to identify the interlocutor and to identify who wants to talk. We believe at the potential of RECOLLVE to support awareness communication among many different users. The main idea is shown in Figure 9. The context refers to a virtual classroom where there are two roles: teacher and student. Teacher and students are represented by avatars. When the *teacher* transfers the "microphone" to the student which holds his arm up (student_3), the Integration-Agent, according to the established rules, sends a message asking LEICA to turn the microphone of that student ON, giving him the permission to talk. This example illustrates the social protocol mastering the collaborative work. The teacher can take the microphone at any moment, on the other hand, the students cannot do it without asking (holding their arms up) to teachers

Note that the problems related to the audio/video communication, are handled by the Skype system. RECOLLVE is just responsible for the scene synchronization: the avatar positions, for example.

6 Perspectives for Future Work

In this article, we proposed RECOLLVE - a collaborative virtual environment agent-based that provides mechanisms to users to represent any kind of colaborative applications. This is possible thanks to event and interaction model defined. Moreover, we defined an API to collaborate beyond a virtual scene. We made also a first integration experience to represent a collaborative navigation session inside a virtual scene. For future works, we intend to analyze and evaluate the usability of this virtual interface.

References

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

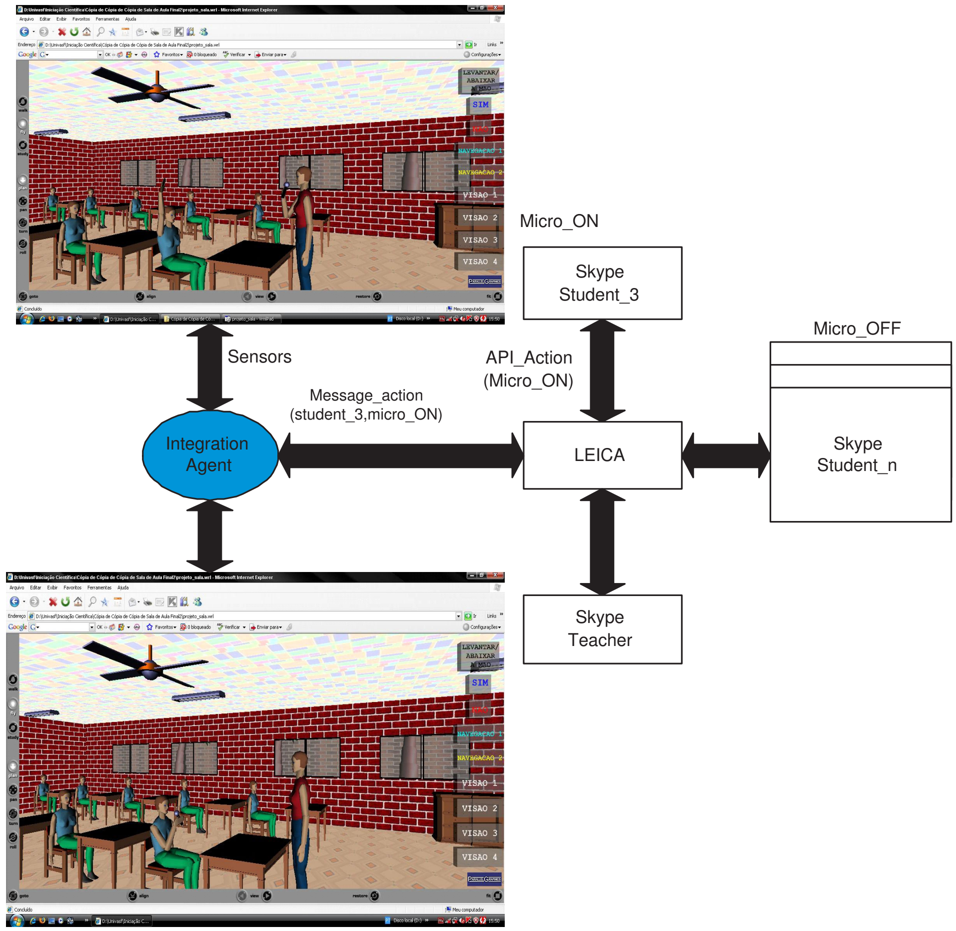

Figure 9. Representing into the virtual scene an audioconference session

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 Jesús 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] I. Project, "Web site: http://invite.fh-joanneum.at/, may 2008."

[6] 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.

[7] P. Dax, "Virtual Reality Engine - web site: http://vreng.enst.fr/html/index.html."

[8] VNET, "Web site: http://www.csclub.uwaterloo.ca/u/sfwhite/vnet."

[9] 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.

[10] 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.

[11] 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.

[12] N. Richard, “InViWo agents: Write once, display everywhere,” in *Proceedings of the Web3D Conference*, 2003.

[13] 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.

[14] 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.

[15] “Web site: http://jade.tilab.com/ jul, 2008.”

[16] “Web site: http://www.fipa.org/ jul, 2008.”

[17] 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.

[18] 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.

[19] 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.

[20] J. V. dos Santos Filho, R. L. Gomes, A. de Castro Pinto Pedroza, and J.-P. Courtiat, “RECOLLVE - REpresenting COLLaboration in Virtual Environments,” in *Proceedings of CGVR’08 in WORLDCOMP’08*, 2008.

[21] R. L. Gomes, “Leica: Un environnement faiblement couplé pour l’intégration d’application collaboratives,” Ph.D. dissertation, Université Paul Sabatier - Toulouse - France, 2006.

[22] G. de Jesús Hoyos Rivera, “Colab: Conception et mise en oeuvre d’un outil pour la navigation coopérative sur le web,” Ph.D. dissertation, Université Paul Sabatier - Toulouse - France, 2005.