

Coordinating Collaborative Work with RECOLLVE

Santos Fh, J. V., Gomes, R. L., Bacurau, R. M. , Pedroza, A. de C. P. and Courtiat, J.-P.

CECOMP/UNIVASF

Juazeiro, BA - BRAZIL

{jose.valentim,rodrigo.bacurau}@univasf.edu.br

DINF/UFES

Vitria, ES - BRAZIL

rgomes@inf.ufes.br

GTA/COPPE-Poli/UFRJ

Rio de Janeiro, RJ - BRAZIL

aloycio@gta.ufrj.br

LAAS/CNRS

Toulouse, FRANCE

courtia@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 among users geographically distant, many CSCW (Computer Supported Cooperative Work) systems like audio and video-conference, shared editors, shared white-board 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 detailed in this article the potential of RECOLLVE to represent and to coordinate into the virtual scene a very large set of collaboration activities, including their social aspects.

KEYWORDS: collaboration, virtual environments, awareness, coordination.

1. INTRODUCTION AND RELATED WORK

Users geographically distant need to exchange information, to communicate and to cooperate. This context creates a demand for collaborative applications. The number of collaborative tools (CSCW systems) has increased over the past years. Currently, many collaborative applications are available for users, such as audio-conference and video-conference tools, shared white-board, shared text

editors, etc. The challenge is to make these tools work together to support real cooperative work. 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]. In this context, the integration tools play a very important role, because they allow to integrate different CSCW systems to support collaborative work. This fact explains the increasing number of systems to integrate collaborative applications, such as the LEICA system [3], for example. This kind of system provides users with a great flexibility to establish collaborative sessions, allowing them to customize tools according to their needs and preferences.

However, one of the most critical problems on the deployment of cooperative work systems is exactly to provide users with a real perception of presence (awareness) of any user he or she cooperates with. Awareness represents the common conscience of a work group that knows the general context of a given work. This group conscience corresponds to the understanding a user has about: i) which whom he works with; ii) what each user does; how do users interacts [4]. Awareness of a work group allow to define the context of this work. In a collaborative system, many people can do work simultaneously on the same artifacts. In the shared work, the actors do not know actions of the other user. Thus, the awareness of individual and group activities become a critical information to make possible the collaboration successful.

The awareness presents the follow advantages:

- reduced effort to coordinate;
- anticipate users actions;
- help the people to take part in the group activities;
- awareness group and their availability to realize cooperative work;
- awareness of actions realized by the members of the work group;
- awareness of users actions consequence.

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”. Collaborative virtual environments have been used in many applications type, such as: e-learning, training, war simulations, etc. *INVITE project* [5], *DIVE* [6], *EVE*[7], Soares et al. [8], *ADVICE* [9], *Second Life* [10], are examples of CVEs where users can share a 3D environment and interact with other users and objects.

We are interested at support awereness of remote users whom are interested in collaborating. For attain this aim, we consider the requirements below:

- ability to represent the dynamic of collaborative work;
- ability to manager the external collaborative applications;
- ability to represent the users access rights;
- roles support;
- open integration of CVE to others collaborative tools.

The Table shows a comparative among many CVEs. The most of them support the role concept. This concept is very significant because in collaborative work, often, people play different functions with different responsibilities and different access rights. As regards to user rights, no one of the CVEs studied consider to represent this concept

into the virtual scene. This concept is very important because it makes stronger the representation of social rules.

All CVEs studied provide some degree of ability to represent collaborative work. *INVITE* system is a platform for tele-learning that provides a shared white-board, supports multi-modal interaction among users by text, video and audio communication.

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.

Soares et al. [8], propose a 2D-3D hybrid interface. User actions on 2D application shared interface in a populated virtual world are represented in the 3D space.

Second Life [10] provides their users with very detailed and advanced avatars that are often also extremely customizable. Users choose among many virtual worlds to navigate and interact with other users through a chat system. However, the avatar-object interaction possibilities are limited to a reduced set of predefined actions.

As regards to the capacity to integrate external applications, *DIVE* provides a very restricted integration with other collaborative applications. The other CVEs do not provide this feature, as consequence, the ability to manager external collaborative tools is not supported by these CVEs. Thus, their possibilities of collaboration are restricted to the CVE itself.

We also found intelligent virtual environments, agent-based systems, with focus on different themes, like Education, Entertainment, Training, Health, etc.

The Virtual Classroom [11] use Intelligent Virtual Environments to support therapy to children with attention-deficit/hyperactivity disorder.

Panayiotopoulos et al proposed at [12] an interactive application that guides the user inside a virtual university. Visitors can communicate with the program through a command driven system and it has a virtual representation of

Table 1. Comparative table among CVEs

	Roles	Integration	Access Rights	Represent Collaborative Work	Manager Collaborative Tools
INVITE	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>No</i>
DIVE	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>No</i>
EVE	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>No</i>
Soares et al	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>No</i>
ADVICE	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>No</i>
Second Life	<i>No</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>No</i>

their requests.

STEVE (*Soar Training Expert for Virtual Environments*)[13] is a virtual environment to teach naval procedures. In this environment, there is an animated agent that teaches to users how to machines and motors work. STEVE shows to users the tasks to be executed and it explain its actions, moreover, it evaluates the performance of the users.

All of them have interesting characteristics, but no one focuses on representing collaborative activities. In [14], Santos et al. shown the potential of RECOLLVE to support collaborative applications such as E-learning, Cooperative Browsing and Virtual Office. In this article, we stressed the potential of RECOLLVE to coordinate the collaborative work in a general way.

RECOLLVE provides a high degree of ability to represent collaborative work because each object has associated to it a functionality set knowledge by the avatars. Then, the avatars realize actions on the objects respecting their functionality. Moreover, RECOLLVE supports also the role concept and the mechanism to control the user access rights is object-centered, making easier to represent into the virtual scene the access politics defined in the collaborative session. Another important feature present at RECOLLVE, is its API that allow to control external collaborative tools, increasing thus the collaboration possibilities.

2. RECOLLVE - THE ARCHITECTURE

The aim of RECOLLVE is to serve as a virtual platform to represent the dynamic of collaborative tasks. Therefore, is extremely important to manage user roles and the access rights associated to them. We are not engaged in defining user access polices, but at representing them into the virtual scene. This aspect is very important because its aim is to transcribe to the virtual scene a natural aspect of real sys-

tems: the social protocol mastering the real collaborative work.

User rights are represented in the virtual scene by handling 3D objects. A user could handle an object depending on its role, its scene position, or another constraint defined. Regarding to the roles, users playing different roles could have different access level to a same object, like happens in real life. For example, a student cannot sit down at the teacher chair. This is a social protocol. The aim is to represent into RECOLLVE the dynamic of a collaborative activity including their social protocols. In RECOLLVE, the access control is object-centered, i.e, the object itself defines if a user can handle it or not and what actions he or she can make on it. The idea is associate to each object a set of access rules. This idea made us to choose the Agent Paradigm to model the RECOLLVE system.

The great advantage of using Agents is its capacity to take decisions from a user interaction with its environment, by sensors, and to make an action based on the collected data.

RECOLLVE is an agent-based CVE focused on representing collaborative applications in virtual environments. Figure 1 shows its multi-agent architecture. In RECOLLVE, all entities are represented by agents which communicate and interact among themselves. Objects and avatars are handled in the same way due to the communication facilities among agents provided by standard languages.

We use the JADE [15] platform 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 into the ACL language defined by the FIPA - Foundation for Intelligent Physical Agents - [16] an international standard for agent interoperability.

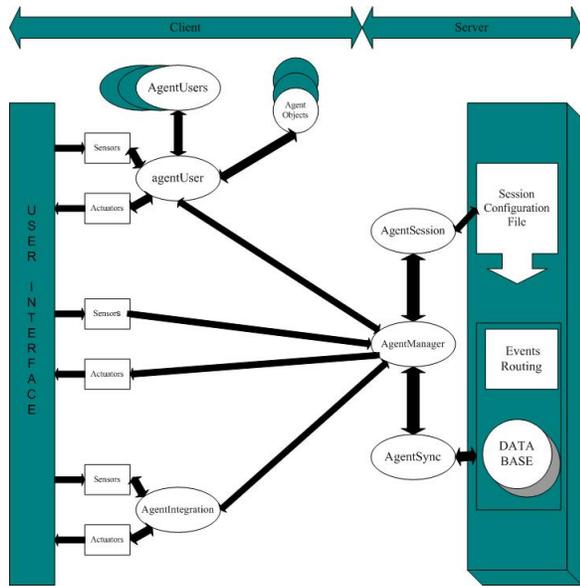


Figure 1. Multi-Agent Architecture of RECOLLVE

We have defined six types of agents:

- user-agent: users are represented into the virtual scene by embodied *User-Agents*, referred as avatars. The *User-Agent* model is very simple: when an avatar is coming near to an object or 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.
- object-agent: objects are represented into the virtual scene by embodied *Object-Agent*. The architecture of Object-Agent is shown in Figure 2. Objects define what actions an avatar can realize on it, based on early defined rules. Figure 3 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 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. 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 user 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.

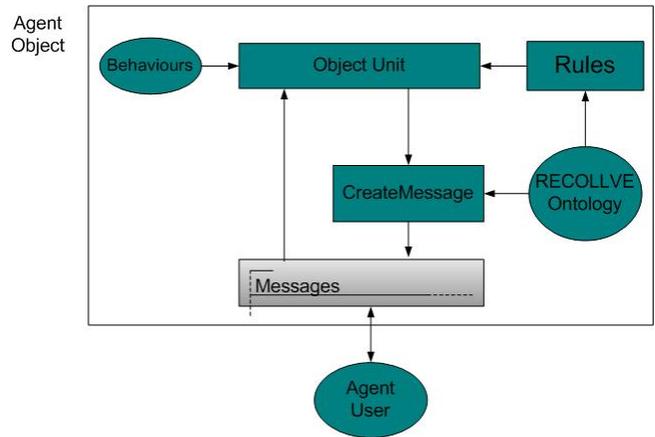


Figure 2. Architecture of Agent-Object

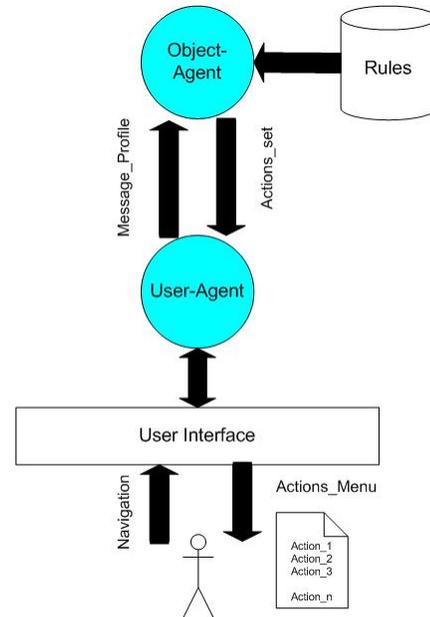


Figure 3. Agent-user and Agent-object interaction

- manager-agent: the Manager-agent coordinates actions of other agents; it sniffs actions from users and from their agents and it also keeps the virtual scene synchronized among all users. Moreover, it makes actions due to user 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 misbehavior user. The architecture of Manager-Agent is shown in Figure 4.
- integration-agent: serves requests from external applications to realize actions inside a virtual scene and

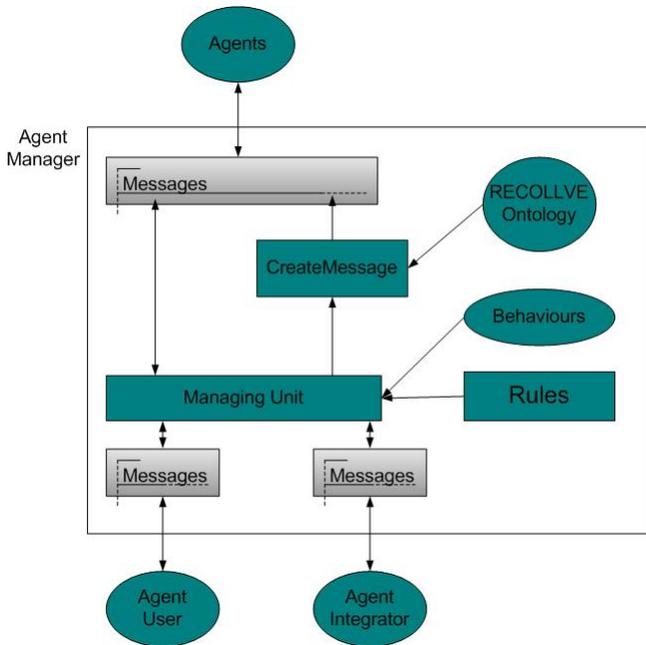


Figure 4. The Manager-Agent

vice-versé. When it receives a real event notification message from the *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 a virtual event notification message from the Manager-Agent, the Integration-Agent asks the Integration-Environment to make a specific action. Figure 5 shows the interactions of the Integration-Agent with the Agent-Manager and the Integration-Environment.

- sync-agent: it synchronizes the virtual scene. The Agent-Sync is responsible for keeping the virtual scene synchronized in all clients.
- session-agent: it controls the collaborative virtual session. The Agent-Session is responsible for adding new users to the virtual session and for removing users that quit the virtual session. Moreover, it validates users login and password.

2.1. AN ONTOLOGY EXAMPLE

An application multi-agent Ontology is defined as a concepts, actions and predicate set. This definition is need

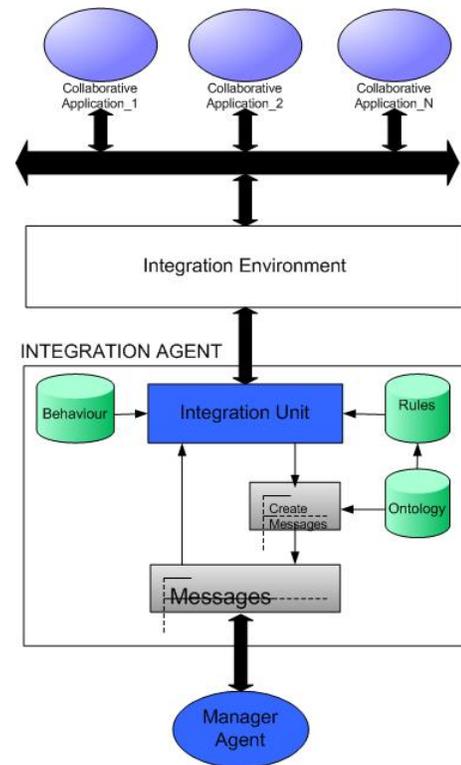


Figure 5. The Integration-Agent

because it is fundamental that there is among agents the common understood about the domain objects. Thus, such information are interpreted in a same way for all agents in the society [17]. Using Ontology's, we can describe relationships between objects, including semantics information.

Figure 6 shows the concepts used by the agents to an E-Learning application.

The *RoomClass* concept is associated to a virtual class room. Each virtual class room is composed by a *VirtualEnvironment* that corresponds to the VRML code and different classes of objects, such as:

- ObjectRoom: these objects are used to compose the environment. The agents realize suitable actions on these objects, but they have not a semantic at the collaboration context. For example, a chair, a table;
- ObjectToken: These objects have a strong semantic representation at the collaboration context. They identify the interlocutors at the collaboration activities. For example, a microphone, a floor, a pilot;

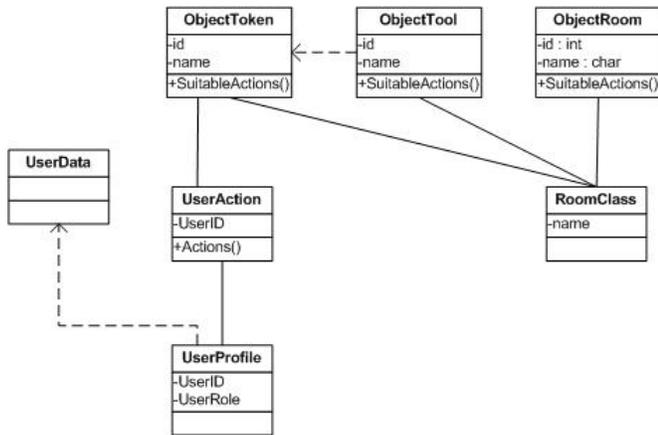


Figure 6. An Ontology for an E-Learning scenario

- **ObjectTool**: these objects are used by agents to represent a collaboration activity. These objects have also a strong semantic representation at the collaboration context. For example, a White Board.

Another concept established is the *UserProfile*. This concept associates a user to the role attribute. The concept *UserProfile* is associated to a unique *UserData*, where are stored the data of users. The actions defined in the RECOLLVE

LVE Ontology corresponds to operations agents demand to other agents. Each action have as attributes the information needs to agent execute the task. The Predicates defined

for the RECOLLVE Ontology are used to fill some needs to specify the communication between agents. Predicates are used in messages sent by the UserAgent the ObjectAgent asking the actions set available. At JADE platform,

the content of each message is a reference expression described at Semantic Language (SL)[16] specified by FIPA, which asks what object satisfy the predicate. The UserAgent

build a reference expression in SL that represents the follow question: "what the actions set available to me, considering that my role is X and i own the object Y?". This relationship among *ActionsSet*, *Role* and *OwnObject* is represented by the predicate *IsObjectAvailable*.

3. IMPLEMENTATION ISSUES

The first version of RECOLLVE prototype has a client/server architecture as shown in Figure 7. RECOLLVE uses the *VIP - VRML Interchange Protocol* [18] 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 browser¹. 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 Manager-Agent

executes on the Server. 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.

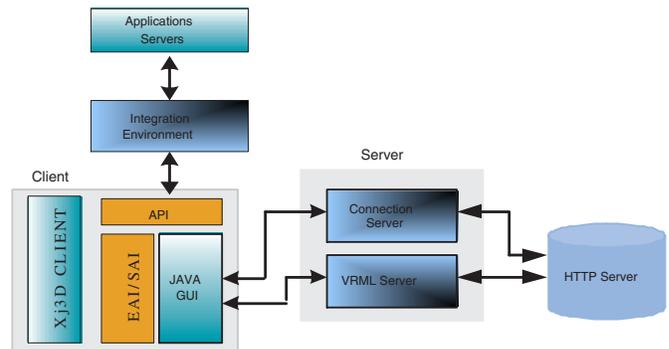


Figure 7. RECOLLVE Implementation Architecture

3.1. API

In [19], *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, external events have consequences in the virtual scene. In the same way,

¹<http://www.xj3d.org>

events in the virtual scene have consequences in external applications. For example, the avatar selection of another user could start automatically an audio-conference 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.

4. SUPPORTING AWARENESS IN COLLABORATIVE APPLICATIONS

The aim of this article is to show the potential of RECOLLVE as a virtual interface to coordinate collaborative work, mapping real actions into virtual actions, or mapping user actions into avatar actions. A virtual interface is more intuitive and makes awareness stronger, making the communication and the relationship among users easier.

In the context of CSCW systems, the audio-conference tools allow users to talk with many people simultaneously; the transmission quality play a very important role for the communication understood. Moreover, it does not consume an enormous bandwidth. The audio is one of the most riches communication media, but using audio-conference with many people simultaneously as the only communication media could present difficulties to identify the interlocutors.

In the context of *E-learning*, we can imagine a Virtual University, where students could share an entertainment space and where there is a course at each virtual class. Once the teacher enter into a virtual class, the Integration-Agent ask LEICA [3] to starts the audio-conference. LEICA is a rule-based system of integration where applications (CVE and Skype in this example) interact by message event notifications.

Symbolizing the audio-conference session, the Integration-Agent makes a “Microphone” to appear in the virtual scene, as shown in Figure 8. Teacher and students talk through an audio-conference tool, like the Skype system, for example. However, the main problem with audio-conference systems is to coordinate the communication when there are many users participating: to identify the interlocutor and to identify who wants to talk.

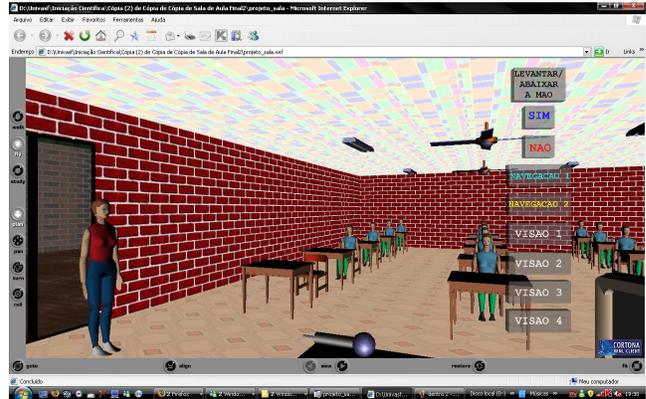


Figure 8. The microphones symbolizes the start of an audio-conference session

We believe in 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. In the first scene, the *teacher* keeps the “microphone” symbolizing that he has the *token* to talk. In this situation, only the microphone of the teacher is turned ON. The microphones of all other users are turned OFF. Thus, we assure that no *student* will interrupt the *teacher* without permission. If a student has any question, he should keep his avatar arm up. When the *teacher* transfers the “microphone” to the *student* which holds his avatar arm up (student_3), the Integration-Agent, according to established rules, sends a message asking LEICA to turn the microphone of that student ON, giving him 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) the teacher. Another important feature present at RECOLLVE is to allow users to manager the external collaborative tools. For example, still in the virtual classroom, if the teacher divides the class in two or more groups to do a specific work; in this when the avatars arrive at the table, the

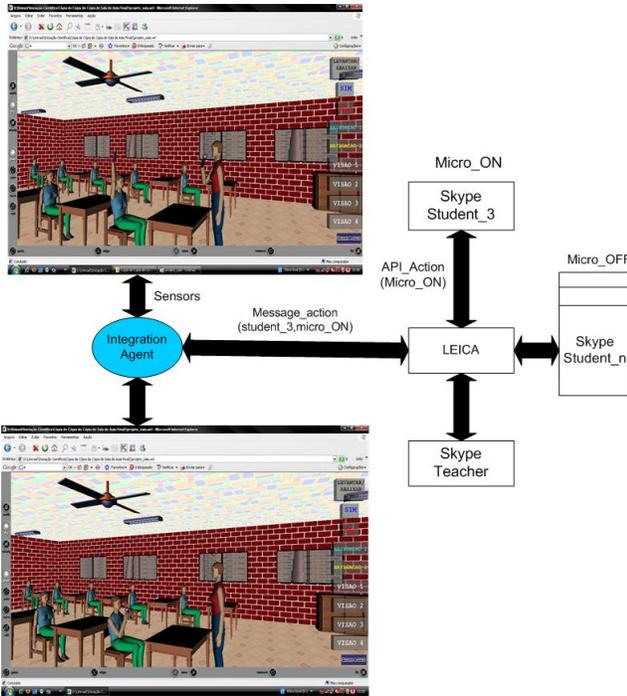


Figure 9. Representing into the virtual scene an audio-conference session



Figure 10. Representing two audio-conference groups into the virtual scene

Integration-Agent ask LEICA to create at Skype, two or more audio-conference groups. Then, each group will have a "Microphone" to control their conversation. This situation is shown in Figure 10. This feature makes RECOLLVE flexible enough to explore the functionality of the external applications.

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.

Another experience, shown in Figure 11, is to represent a Collaborative Browsing session using COLAB systems. COLAB is a Web collaborative browsing system. 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 users: to browse together in the Web (a user drives the browsing) or users to separately browse in the Web, but having awareness of the other browsing users.

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 allows 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, the 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 11 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 the other user in COLAB too.

A virtual classroom could be also an application scenario. The teacher wants to show a WebSite to the students to explain a content, or he wants to show an experience. Then, the teacher can navigate through the Web and the users get the same URL which is shown into the virtual room. The avatar representing the teacher takes the token,

symbolizing that he has the control at that moment. If a student has any question, he should keep his avatar arm up. When the *teacher* transfers the "floor" to the *student* which holds his avatar arm up, the Integration-Agent, according to established rules, sends a message asking LEICA to notify that a student is the leader now, and the other students and the teacher will get the same URL. The teacher can at any moment take the "floor" again.

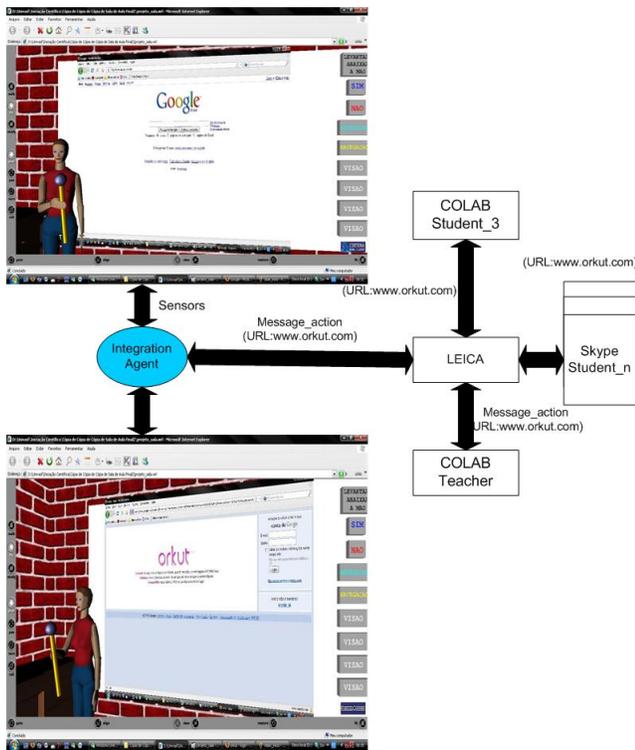


Figure 11. Representing a virtual scene in a Colab session

The second scenario of integration is a Virtual Office shown in Figure 12. Here again, three collaborative applications, under development - our RECOLLVE prototype, a Chat tool and a Shared Editing - are integrated by the LEICA [3] system.

In Chat systems, users share experiences, opinions and knowledge by exchanging messages. Representing this kind of application into a virtual scene, makes the awareness among users stronger. Figure 13 shows a representation of a Chat system scenario in a virtual scene.

Shared Editing is a kind of tool that allows two or more users to edit together a shared document. The main problem on the deployment of this kind of system is to manager concurrent tasks, when two or more users simultaneously edit the same document. Figure 13 shows, in the first scene, one scenario where three users, represented by

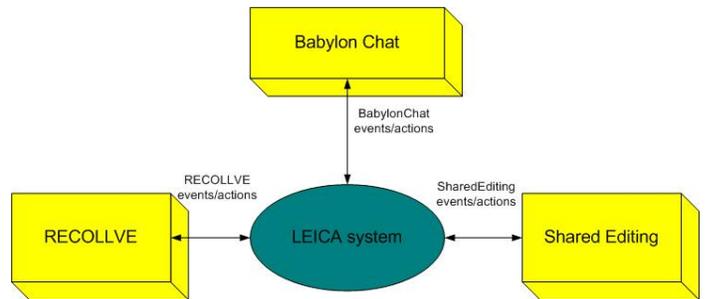


Figure 12. An E-Learning scenario

their avatars, handle three different parts of the document. Meanwhile, other two avatars wait. The second scene shows that the yellow avatar finished to handle the part number two of the document and it waits now for the part number three. The other users could see that the part number two is now free.

This situation make stronger awareness of who are handling what part of the Shared Editor. Obviously, the number of parts (chapter, page, etc.) the Shared Editor allows to handle depends of its granularity.

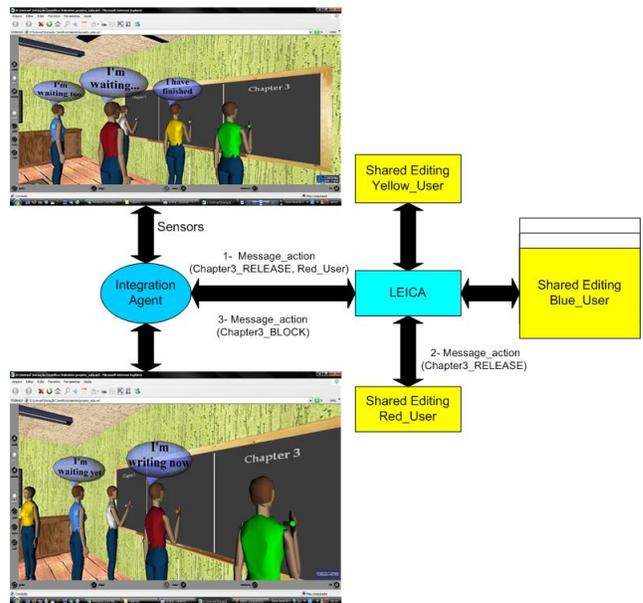


Figure 13. Representing into the virtual scene a Shared Editing session

When the yellow avatar finishes to handle the part number two of the document, the Integration-Agent, according to predefined rules, sends a message asking LEICA to release the user access to the *Red_User*, whose avatar starts the queue, to that part of the document. The part number two

of the document remains blocked for other users and it only will be released by the *Red_User*.

These scenarios illustrated above show the potential of RECOLLVE to be used as a platform for many kind of collaborative work: *E-Learning*, entertainment, virtual office, entertainment or training, among others.

5. PERSPECTIVES FOR FUTURE WORK

In this article, we presented the potential of RECOLLVE, a collaborative virtual environment agent-based, to represent any kind of collaborative applications. We realized also a first integration experience to represent an audio-conference session inside the virtual scene, in the context of E-Learning application. For future works, we intend analyze methodologies to evaluate the usability of this virtual interface. A performance analysis of the system is also needed.

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] 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.
- [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] C. Bouras and T. Tsiatsos, "Distributed virtual reality: building a multi-user layer for the eve Platform," in *Journal of Network and Computer Applications*, vol. 27. Elsevier, 2004, pp. 91–111.
- [8] 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.
- [9] "ADVICE: Um ambiente virtual colaborativo para o ensino a distancia," Jun. 2004.
- [10] "Second life: Your World, Your Imagination, World Wide Web, <http://secondlife.com>, may 2008," December.
- [11] A. A. Rizzo, T. Bowerly, G. BuckWalter, D. Climchuck, R. Mitura, and T. D. Parsons, "A virtual reality scenario for all seasons." Kluwer Academic Press, 1999, pp. 33–42.
- [12] T. Panayiotopoulos, N. Zacharis, and S. Vosinakis, "Intelligent Guidance in a Virtual University," in *CNS Spectr*, vol. 11, 2006, pp. 35–44.
- [13] W. L. Johnson, "Pedagogical Agent Research at CARTE," in *AAAI*, vol. 22. Winter, 2001, pp. 81–94.
- [14] 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.
- [15] "Web site: <http://jade.tilab.com/> jul, 2008."
- [16] "Web site: <http://www.fipa.org/> jul, 2008."
- [17] 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.
- [18] VNET, "Web site: <http://www.csclub.uwaterloo.ca/u/sfwhite/vnet>."
- [19] R. L. Gomes, "Leica: Un environnement faiblement couplé pour l'intégration d'applications collaboratives," Ph.D. dissertation, Université Paul Sabatier - Toulouse - France, 2006.