A NEW INTERFACE FOR COLLABORATIVE TELEOPERATION

Narjes Khezami, Samir Otmane and Malik Mallem

Complex System Lab, University of Evry
{khezami, otmane, mallem}@iup.univ-evry.fr

Abstract: This paper presents a conceptual architecture and a new interface developed for collaborative teleoperation via Internet. This interface is based on a MAS (Multi-Agent System) devoted to collaboration. The MAS formalism is based on a new approach combining astutely two research area: MAS and CSCW (Computer Supported Collaborative Work). In the first area, a formal model of the MAS (Ferber approach) is used to give us features of the agent. The second area provides us a useful approach of collaboration based on the communication, coordination and production theories.

Keywords: Interface, Agents, Teleoperation, Telerobotics, Co-operation.

1. INTRODUCTION

The World Wide Web (WWW) has become today an important means to bring robots to the public. Several robot systems, which can be operated over the WWW, have been developed over the last few years (Hu, *et al.*, 2001; Safaric, *et al*, 2001; Otmane, *et al.*, 2000; Nasa, 2004). On the other hand, many current teleoperation architectures allow Human Operator (HO) to perform remote complex tasks and let multi-user interface for only collaborative control of the robot (Goldberg, *et al.*, 2000; Goldberg, *et al.*, 2002).

In this paper we present an interface of collaboration based on MAS devoted to collaborative teleoperation architectures. This interface uses a MAS formalism which is based on a new approach combining astutely two research area: MAS and CSCW (Laurillau, 2002; Ellis, *et al.*, 1991; Salber, 1995). In the first area, a formal model of the MAS (Ferber approach) (Ferber, 1997) is used to give us features of the agent. The second area provides us a useful approach of collaboration based on the communication, coordination and production theories.

The MAS of collaboration is integrated in the kernel of ARITI (Augmented Reality Interface for Telerobotic applications via Internet (Otmane, *et al.*, 2000; Otmane, *et al.*, 2002; Nasa, 2004)) in order to make it multi-user, i.e. allow several users to collaborate and control the robot at the same time. Consequently the system of collaboration manages the communication between users, conflicts of task execution as well as the different user behaviour. ARITI is a teleoperation system which allows only one user to accomplish a task of teleoperation. It allows first connected to control a robot via Internet, this robot is located in our research laboratory. If a second user connects, he is only allowed to supervise the first one during his manipulation, or to prepare a teleoperation mission on simulation.

In the section 2, modelling of collaborator agent is presented. The section 3 is devoted for the architecture of the interface of the collaborative multi-agent system. Some conclusions and futures works are presented in the section 4.

2. MODELLING OF AGENT COLLABORATOR

2.1. Collaboration specification

The CSCW is the field of study that examines how technology affects group interaction, and how technology can best be designed and built to facilitate group work. Activities in that domain are known by the notions of groupware. Ellis (Ellis, *et al.*, 1991) defines groupware as “computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment.”

To structure groupware analysis and to propose a means of classification of the many tools permitting to reach the CSCW objectives, functionalities of the cooperative tools are divided in the functional clover as shown in Fig.1 (Laurillau, 2002; Salber, 1995). Then a groupware application covers three kinds of services: production, communication and coordination.

- The communication space refers to person-to-person communication such as e-mail, relay chat, media space.
- The coordination space covers activities dependencies including temporal relationships between the multi-user activities. It also refers to the relationships between actors and activities.
- The production space refers to the objects produced by a group activity or to the objects shared by multiple users.

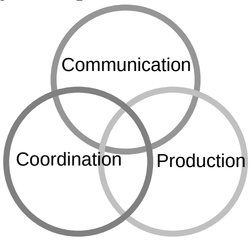

Fig. 1 Clover's Model

The importance of each of the three spaces depends on the nature of tools considered. So, the distinction according to the three spaces isn't strict because it's possible that a coordination activity takes place after a communication activity. For example, to make an appointment with someone on the phone is a coordination act based on a communication one.

2.2. Formal Modeling of Collaborator Agent

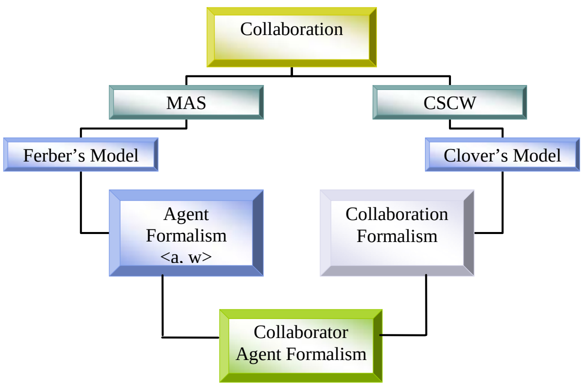

Fig. 2 Collaboration Modeling Process

In this section we present a formal model for the collaborator agent. This model, as shown in Fig. 2, is the combination of the collaboration modeling presented in (Khezami, *et al.*, 2004) and the formal model of SMA presented in (Ferber, 1997) (Ferber's Model.)

A collaborator multi-agent system is a couple $\langle i, w \rangle$, where i is a collaborator agent and w is the world in which the agent evolve (Eq. 1).

$$i = \langle P_i, Percept_i, F_i, Infl_i \rangle$$
$$w = \langle \mathrm{E}, \Gamma, \Sigma, R \rangle \qquad (1)$$

Where,

- E is the space in which the agent evolves. It's represented by agents in the environment.
- Γ is the set of agent actions, which modify the world evolution. For us, Γ is the set formed by the three subsets $\{Inf_{ij}\}, \{Actions_i\}, \{Results_i\}$ such as (Eq. 2):

$$\Gamma = \{\{Inf_{ij}\}, \{Actions_i\}, \{Results_i\}\} \qquad (2)$$

- Σ is the set of agent states. For us a collaborator agent i has three states: communicate, coordinate or produce ($comm, coor, prod$). As presented in Eq. 3.

$$\Sigma = \{comm, coor, prod\} \qquad (3)$$

- $Percept_i$ is a set of stimuli and sensations that the composed of three subsets $\{Inf_{ji}\}, \{Actions_j\}, \{Results_j\}$ such as (Eq. 4):

$$Percept_i = \{\{Inf_{ji}\}, \{Actions_j\}, \{Results_j\}\} \qquad (4)$$

- P_i is an agent perception function (Eq. 5.)

$$P_i : \{comm_i, coor_i, prod_i\} \rightarrow \{\{Inf_{ji}\}, \{Actions_j\}, \{Results_j\}\} \qquad (5)$$

- F_i is the agent behaviour function that determines the agent's state from its perceptions and its previous state (Eq. 6.)

$$F_i : \{comm_i, coor_i, prod_i\} \times \{\{Inf_{ji}\}, \{Actions_j\}, \{Results_j\}\} \rightarrow \{comm_i, coor_i, prod_i\} \qquad (6)$$

- $Infl_i$ the agent action function, which modifies the evolution of the world while producing influences (Eq. 7.)

$$Infl_i : \{comm_i, coor_i, prod_i\} \rightarrow \{\{Inf_{ij}\}, \{Actions_i\}, \{Results_i\}\} \qquad (7)$$

- R the law of evolution of the world (Eq. 8.)

$$R : \{comm_i, coor_i, prod_i\} \times \{\{Inf_{ij}\}, \{Actions_i\}, \{Results_i\}\} \rightarrow \{comm_i, coor_i, prod_i\} \qquad (8)$$

2.3. Conceptual Modelling of Collaborator agent

In this section we present the conceptual model of our collaborator multi-agent system.

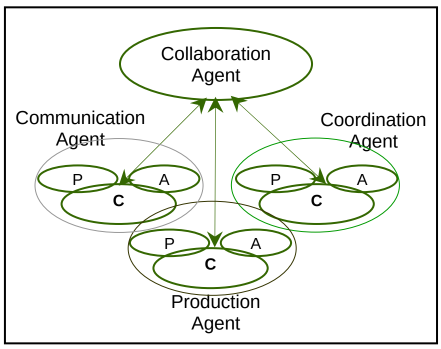

Fig. 3 Collaborator Agent

Our conceptual architecture is composed of PAC* agents (Calvary, *et al.*, 1997; Nigay, *et al.*, 1997). In which, every collaborator agent is composed of three agents dedicated respectively to the *communication*, the *coordination* and the *production* (Fig. 3). These agents are bound to an agent *ciment*, that we call agent *collaboration*. This last one ensures the communication, on one hand, between the three dedicated agents and on the other hand, between the collaborator agent and the other collaborator agents. In this architecture model, a collaborator agent interact with the other collaborator agents of the system through the intermediary of its agent collaboration and its dedicated agents (communication, coordination, production) except that every one of them communicate always with the corresponding agent of other collaborator agents.

The collaboration agent ensures two principal functions; it allows communication between the three dedicated agents inside the collaborator agent; and it establishes a direct interaction between every dedicated agent and its corresponding one of other collaborator agents in order to communicate together without the help of agent collaboration. This direct communication between the three dedicated agents, inside a collaborator agent, and their corresponding agents of the other collaborator agents, allow to increase the efficiency of the system because the communication don’t depend on the collaboration agent, so that avoid the “bottle neck” effect made by centralization.

3. ARCHITECTURE OF MULTI-AGENT SYSTEM FOR COLLABORATION

3.1. Collaboration process

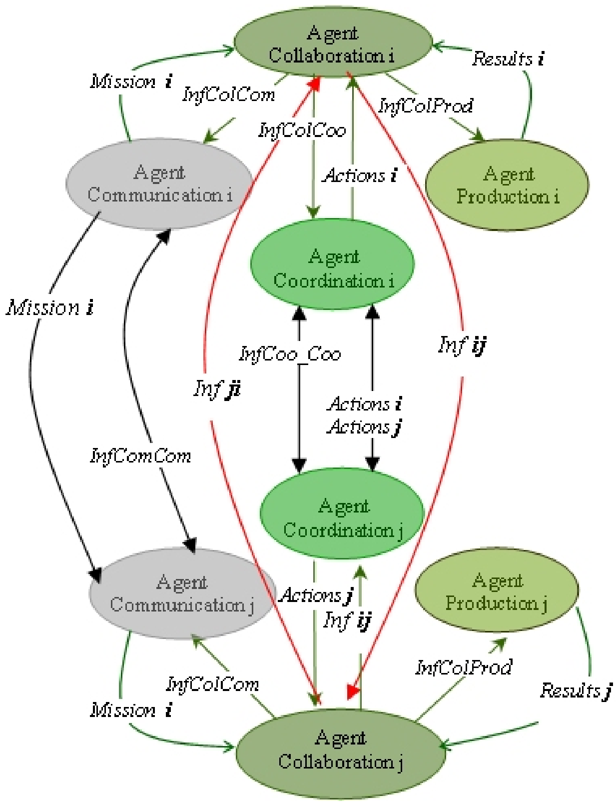

Fig. 5 Collaboration process between two agents

The Fig. 5 shows the collaboration process between two collaborator agents (agent collaborator *i* and agent collaborator *j*.) The agent collaborator *i* is the agent that starts collaboration process. All information between agents are represented by the *Inf. Mission$_i$* is the mission chosen by the agent communication *i* and the agent communication *j*. *Actions$_i$* and *Actions$_j$* are actions defined by agent coordination *i* and agent coordination *j*. *Resluts$_i$* and *Results$_j$* are the confirmation of execution of the agents production *i* and *j*.

3.2. Implementation architecture

The collaboration model is implemented using the JADE framework (Java Agent DEvelopment framework (Jade, 2004)). JADE is a software development framework aimed at developing multi-agent systems and applications conforming to FIPA standards for intelligent agents. JADE is written in Java language and is made of various Java packages.

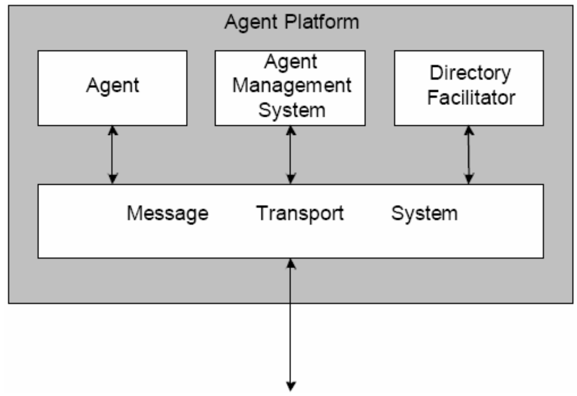

Fig. 6 Reference architecture of a FIPA Agent Platform

The standard model of an agent platform, as defined by FIPA, is represented in the Fig. 6. The Agent Management System (AMS) is the agent who exerts supervisory control over access to and use of the Agent Platform. Only one AMS will exist in a single platform. The AMS provides white-page and life-cycle service, maintaining a directory of agent identifiers (AID) and agent state. Each agent must register with an AMS in order to get a valid AID. The Directory Facilitator (DF) is the agent who provides the default yellow page service in the platform. The Message Transport System, also called Agent Communication Channel (ACC), is the software component controlling all the exchange of messages within the platform, including messages to/from remote platforms.

The Agent class represents a common base class for user defined agents. Therefore, from the programmer’s point of view, a JADE agent is simply an instance of a user defined Java class that extends the base Agent class. This implies the inheritance of features to accomplish basic interactions with the agent platform (registration, configuration, remote management, …) and a basic set of methods that can be called to implement the custom behaviour of the agent (e.g. send/receive messages, use standard interaction protocols, register with several domains,…). The computational model of an agent is multitasking, where tasks (or behaviours) are executed concurrently. Each functionality/service provided by an agent should be implemented as one or more behaviours. A scheduler, internal to the base Agent class and hidden to the programmer, automatically manages the scheduling of behaviours. The messages exchanged between agents are comply with the ACL messages defined by the FIPA specifications.

In our system, when a client connects, an agent server, that we call generator agent, create him an agent collaborator. For every agent collaborator (Fig. 3); an agent collaboration, an agent communication, an agent coordination and an agent production are created by the agent generator. Once the four agents are created, the agent generator sends the address of each one from them to the agent collaboration; after that it makes them on wait state, waiting for the action of client. Created agents record themselves by the DF agent.

3.3. Global architecture of collaboration

The multi-agent system of collaboration is integrated in the kernel of ARITI in order to make it multi-user, i.e. allow several users to collaborate and control the robot at the same time. Consequently the system of collaboration manages the communication between users, conflicts of task execution as well as the different user behaviour.

ARITI is based on the client/server architecture. The client is the interface run by a connected user. And two servers: Video server performs image compression and transfer to the client. And Robot server allows to telecontrol the robot. If a user connects to system, the ARITI interface (Fig. 7) is run and if he is the first connected, he can control the real robot. Users can't communicate together and can't know connected persons no else their number.

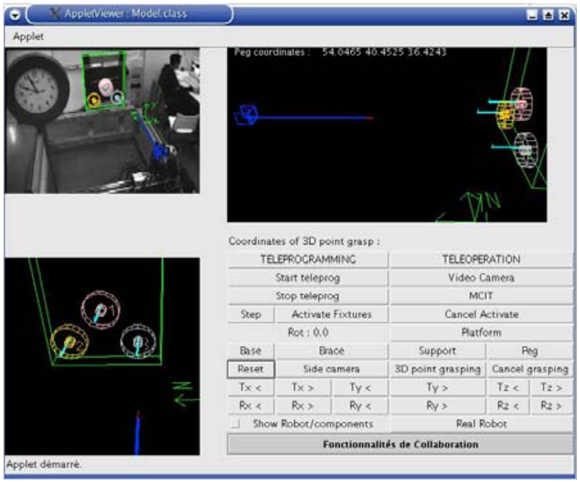

Fig. 7 ARITI Interface

We have integrated the collaborator multi-agent system on the robot server. Now, if user connects to the system, he can collaborate with another one. He can also know the connected persons and their number. And the most important he can share the real robot with the others.

The figure Fig. 8 shows the global architecture of the collaboration system. In this figure there are two connected users and therefore executed two ARITI clients. These users can collaborate together via Internet. They connect directly on the server of MAS, which is bound to its tour to the robot server. Every user will have on his screen the interface of collaboration that contains the list of missions to do and actions to execute for each of missions.

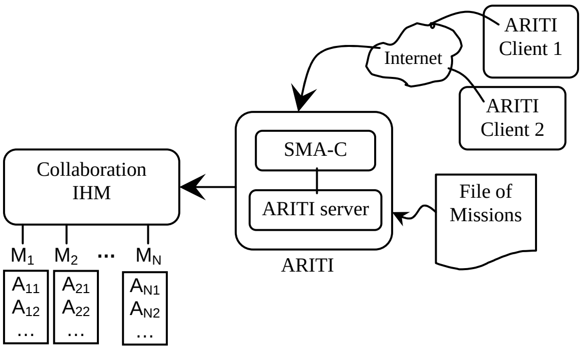

Fig. 8 Global collaboration architecture

For every connected user, the ARITI interface is run. The user chose the collaboration mode then a new collaboration interface is run (Fig. 9). He can communicate with others persons on the system by chatting (public chatting: he chat with all connected users and private chatting: he chose one of them and chat with him on private.) This interface is composed from three windows: the first one is the communication window; the second is the coordination window and the last one is the production window. Each one of those windows communicates with the dedicated agent of the agent collaborator; e.g. the communication window is linked with the agent communication …

Every window is bound directly to an agent of the collaboration MAS. In the window communication we have two sub windows by default: one sub window dedicated to the public chat, a second containing the connected people list. And we can have so many private windows that people with which we want to discuss in private. When a user connects to the system, the server of robot asks him to identify with a login and a password. The robot server transmits the login to the collaboration server that stocks it and displays it in the window of the list of the connected.

In the window coordination we have four sub windows, the first, from the right, contains the list of actions to execute. The second contains the available mission list. The third contains some supplementary

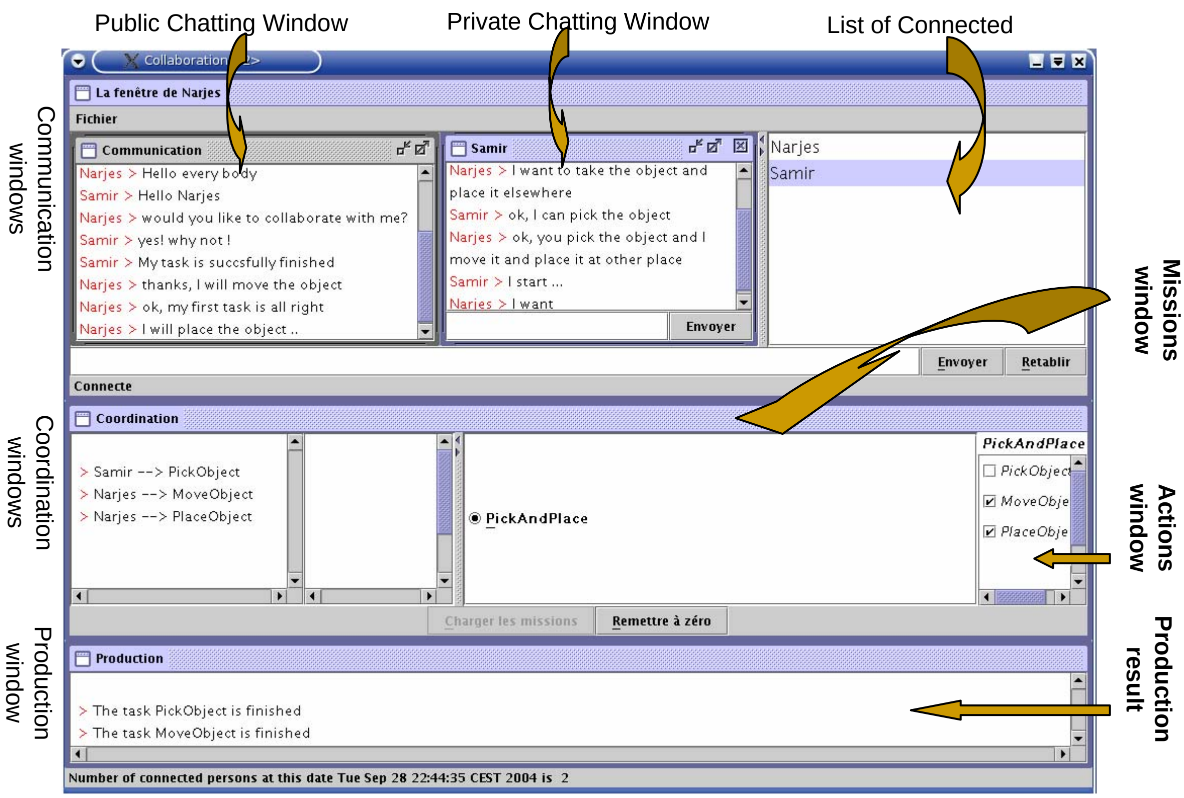

Fig. 9 Collaboration interface

information on missions to do, if it exists some of information. And the fourth contains the list of actions to execute during a task of collaboration with the name of the user that must do this action. The window production contains the result of the execution of the action: if the action is took place well or not.

On the whole when two users reach an agreement to do collaboration task through the window communication, it is the agent communication of each of the two that permits to establish this communication. At the time of loading of missions the agent communication verifies the availability of missions, and then it displays them in the window of missions. Once the two users attain an accord on the doing mission (they can choose at a time only one mission); the agent communication, of each of the two users implied in the communication, transmits the mission chosen to the agent coordination. This last coordinate actions and avoid conflicts between users. Thus, actions are displayed in a list of choice. If a user chooses an action, this last is disabling for the other. Once choices are made, the agents coordination communicates between them and sends each action to execute, one by one and by chronological order, to their agents production. When the agent production receives the action to execute, it verifies that it can control the robot; if it is the case it gives the hand to the user to execute the action.

4. CONCLUSION

A conceptual architecture of collaborative teleoperation is presented with the new interface for multi-user collaboration. This interface is integrated to the ARITI system to give it new features of collaboration.

This interface is a way to allow remote users connected to the Internet to collaborate together for a complex teleoperation task.

The aim of the future work is to use this interface with heterogeneous teleoperation platforms (Fig.10), for example, ARITI system with another system which uses other interactions, perceptions modalities.

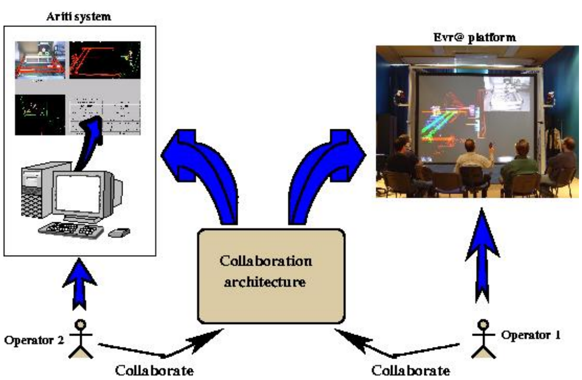

Fig. 10 Heterogeneous teleoperation platforms

REFERENCES

Calvary, G., Coutaz, J. and Nigay, L. "From Single-User Architectural Design to PAC*: a Generic Software Architecture Model for CSCW". Actes de la *conférence ACM Conference on Human Factors and Computing Systems (CHI'97)*, 1997, pages 242-249, ACM Press.

Ellis, Calernce, Gibbs, Simon, Rein and Gail « Groupware :some Issues and experiences ». Journal, *communications of the ACM (CACM)*, 1991, volume 34, number 1, pages 38-58, ACM Press.

Ferber, J. " Les systèmes Multi-Agents: un aperçu général", Techniques et sciences informatiques, Volume 16, 1997.

Goldberg, K., Chen, B., Solomon, R., Bui, S., Farzin, B., Heitler, J., Poon, D. and Smith, G. "Collaborative Teleoperation via the Internet". IEEE International Conference on Robotics and Automation San Francisco, Avril 2000.

Goldberg, K., Song, D., Khor, Y., Pescovitz, D., Levandowski, A. , Himmelstein, J., Shih, J. , Ho, A., Paulos, E. and Donath. J." Collaborative Online Teleoperation with Spatial Dynamic Voting and a Human "Tele-Actor" ", IEEE International Conference on Robotics and Automation, 14 mai 2002.

Hu, H., Yu, L., Tsui, P., and Zhou, Q. Internet-based robotic systems for teleoperation. *Assembly Automation Journal*, 21(2), 2001.

Jade web site, http://sharon.cselt.it/projects/jade/, 2004.

Khezami, N., Otmane, S., Mallem, M., and Ouramdane, N. "A Formal Model for Collaborative Teleoperation". The International Conference on Computing, Communications and Control Technologies (CCCT 2004), 14-17, Août 2004. Texas (USA).

Laurillau, Y. "Conception et réalisation logicielles pour les collecticiels centrées sur l'activité de groupe : le modèle et la plate-forme Clover". Phd. Thesis on Computer Science, University Joseph Fourier, Grenoble, France, September 2002.

Nasa space telerobotics program web site. http://ranier.oact.hq.nasa.gov/telerobotics page/, 2004.

Nigay, L. and Coutaz, J. "Software architecture modelling: Bridging Two Worlds using Ergonomics and Software Properties." Livre, *Formal Methods in Human-Computer Interaction (Palanque et Paterno éditeurs)*, 1997, pages 49-73, Springer Verlag.

Otmane, S., Khezami, N. and Mallem, M. "Distributed Predictive Display for Collaborative Teleoperation on the Internet", in 4th Virtual Reality International Conference (VRIC2002), Pages 111-116, 19-21 june 2002, Laval (France).

Otmane, S., Mallem, M., Kheddar, A. and Chavand, F. "Ariti: an augmented reality interface for teleopeation on the internet". In Advanced Simulation Technologies Conference (ASTC2000) - High Performance Computing, pages 254-261, Wyndham City Center Hotel, Washington, D.C., USA, April 16-20 2000.

Safaric, R., Debevc, M., Parkin, R. and Uran, S. Telerobotics experiments via internet. *IEEE Transactions on Industrial Electronics*, 48(2):424–31, April 2001.

Salber, D. « De l'interaction individuelle aux systèmes multi-utilisateurs: L'exemple de la communication Homme-Homme Médiatisée ». Phd. Thesis on Computer Science, University Joseph Fourier, Grenoble, France, September 1995.