Supporting Team Work in Collaborative Virtual Environments

Gernot Goebbels, Plinio Thomaz Aquino, Martin Göbel
German National Research Center for Information Technology (GMD)
Virtual Environments Group, Germany

Vali Lalioti
Computer Science Department
University of Pretoria, South Africa

Abstract

In this paper we present our approach in creating Collaborative Virtual Environments to provide distributed collaborative teams with a virtual space where they can meet as if face-to-face, coexist and collaborate while sharing and manipulating a set of virtual data in real time. Thereby our approach moves beyond mere integration of video-conferencing and scientific visualization, to create a design framework for CVEs where issues of human-to-computer and human-to-human interaction in projection-based systems are addressed. It focuses on our interaction taxonomy that supports the development of applications which themselves support small groups working together in rear projection-based VEs making use of video conferencing and 6DOF input devices. The approach is exemplified by the design and implementation of a Collaborative Medical Workbench application used for remote education purposes.

Keywords Distributed VEs, Immersive Telepresence, Collaborative Interaction Framework

1 Introduction

The need for high-end collaborative Virtual Environments becomes more pressing due to the globalized nature of today's market. Distributed businesses require support for effective collaboration over distance in order to minimize time and travel costs[2]. Businesses that require high-end visualization of raw data gathered from remote sites [3] [4], as well as remote medical consultation [6] and tele-education, are examples where scientific visualization has been combined with video-conferencing to provide support for collaborative work [8].

In our approach for the design of such an environment, principles developed in the field of human computer interaction and computer supported collaborative work (CSCW) are complemented by techniques

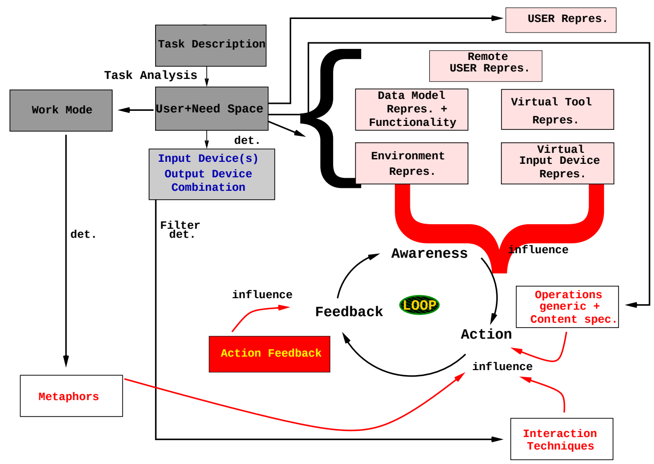

Figure 1: Taxonomy for Autonomous and Distributed, Collaborative Interaction.

for facilitating human to human interaction within a virtual space, for users physically at the same place or for remote collaboration. Design issues involve ways of natural interaction with the virtual data as well as with the remote participants, while preserving shared data consistency.

2 Taxonomy for Distributed, Collaborative Interaction

This approach is a practical tool that can be used as a framework for design and evaluation of VEs. Therefore, we are concerned only with the utility of a taxonomy for these tasks, and not its absolute "correctness". The objective is to facilitate guided design of applications for supporting team work in VEs. One way to verify the generality of the approach is through the process of categorization. Categorization is a good way to understand the low-level makeup of interaction techniques. This categorization may also lead to new design ideas. User tasks need to be specified which

Representation Components	User Representation	# users : x	none	
	Remote User Repr.	# remote users :y	video texture	
	Data Model Repr.	model 1+ model 2 + model 3 +...		
	Data Model Functionality	f(model 1) + f(model 2) + f(model 3) + f(...)	...	...
	Environment Repr.	environment	inventory	
	Virtual Tool Repr.	3D tool cursor 1	3D tool cursor 2	...
	Virtual Input Device Repr.	colored rays	3D cursors	...
	Work Mode	mode 1		
Devices	Input Devices	1.) Stylus / 3 Button Tool	2.) Cubic Mouse / Pinch Glove	
	Output Devices	RWB / RWB	RWB / Wall	RWB / CAVE
Auxiliary	Metaphors	Tug Of War	Work around a table	...
	Operations	generic : grab, zoom, drag, rotate,	content specific : change mode of vis.	
	Interaction Techniques	body-centered	menu	speech recognition
Loops	Action Feedback	highlighting of actions	sonification of actions	...
	Actions	action 1+ action 2 + ...		

Figure 2: The User+Need Space.

will then determine the application and interaction requirements, before the correct VE and interaction techniques can be chosen. This implies a good analysis of the special needs of users the application is going to be designed for.

2.1 User's Task Description and Analysis

Figure 1 shows that the approach starts with a *User's Task description (UTD)*. A task description can look like in the following:
Assume two users who want to connect two virtual wooden laths with each other. They use a hammer and a box of nails. For pulling nails that are wrong pound into the wood they use a pair of pliers. Both stand at either side of a carpenter's workbench. One user holds the wooden laths and the other user pounds the nails with the hammer or uses the pliers respectively.

This description provides information about the number of users involved in the task, the type of material and the tools they use. It describes where the users stand and how they work together. Now a following *User's Task Analysis (UTA)* determines the so-called *User+Need Space (UNS)* which itself is the originator of the flow within the taxonomy graph. This UNS relays the information extracted by the UTA of the UTD. We recommend to do an extensive description and analysis of the user's task in order to find out how the user's need can be satisfied. From our point of view most of the virtual environments lack the addressing of user needs and thus result in a poor user satisfaction and usability.

2.2 The User+Need Space (UNS)

In order to represent the UNS visually we choose an array-like representation. (see Figure 2) However, any other representation form is possible but we think that the mapping between the requirements of the UNS and the features of the Virtual Environment is much more obvious using this type of representation. The first seven features denote representation components (see 2.4). In addition to the number of local and remote users the corresponding representations are included. Although the UNS in Figure 2 is a UNS template, we added different possibilities of realisations. Consequently when working two-handed, different input device combinations are shown, such as a combination of a stylus and a 3 button tool or the combination of a pinch glove and a cubic mouse respectively(see 2.3). These and other combinations are not obligatory, they are just illustrating the usage of the UNS array. Also the items belonging to the operations, metaphors and interaction techniques in the auxiliary section of the array are just of illustrating nature and shows that more than one item can be taken under consideration. Thereby, if in the rows appears an enumeration, the first item or combination has be interpreted as the most appropriate. Then the application designer has to choose one of the suggestions. If there is no enumeration the row represents a list of items that belong together. Then all have to be taken under consideration within the application design.

2.3 Input/Output Device Combination and Working Mode

It is obvious that not all 6DOF input devices for interaction and output devices for interacting can be combined. For example, it is hard to use a Cubic Mouse together with a stylus in a CAVE-like display system if the stylus needs to be used frequently. The reason is simply that for using the Cubic Mouse the user needs both hands which results in putting other input devices away. Combining these input devices with the RWB as output device for example the user has got the possibility of putting unused devices back on the table of the RWB. But of course this cannot be the only reason for choosing a certain type of input/output combination. The selection of the devices is mainly influenced by other factors. Most important fact for the selection of an adequate output device is the amount of users who work together at the same site and of course the size of the data model. The most adequate display system for an architect who shows the pre-visualized interior of a building to the client is a wall or a cylindrical projection and a Cave rather then a RWB or a ReachIn display system. An adequate combination of input devices and output devices has to be found with respect to the user's task and data set of use. Thus input and output combination of interest is directly derivable from the User+Need space as all needs and requirements are already defined there.
The Work Mode is determined by the user's task too. Different modes of work are:

- stand-alone, autonomously and data sets are locally uploaded
- stand-alone, autonomously and data sets are remotely uploaded
- stand-alone, collaboratively and data sets are locally uploaded
- stand-alone, collaboratively and data sets are remotely uploaded
- distributed, collaboratively and data sets are provided by one of the sites, or by a remote (external) data server

The first two items describe the possibility to work alone where data sets are locally available or must be downloaded remotely from a simulation loop for instance. No collaborative working is enabled at all.
The third and the fourth item described collaborative working together using one display system. The data sets are available locally again or have to be downloaded from a remote data server.
The last item is the more interesting one where at least two sites work together. Now the shared data sets can either be provided by one or even more members of the session or be provided by an external data server. The work mode itself it important to determine the metaphors described in 2.6.

2.4 Representation Components

Representation Components denote a very important part of Virtual Environments. They determine how the visual parts in the application are represented. The components are (see Figure 1):

- User Representation
- Remote User Representation
- Data Model Representation and Functionality
- Environment Representation
- Virtual Input Device Representation
- Virtual Tool Representation

As shown in Figure 1 all components except for the *User representation* belong to a group. The User Representation is of interest only to the user and not to the remote partner. Most rear projection-based Virtual Environments do not need an explicit user representation in contrast to HMDs, where the user is typically represented by a hand or a whole body like in *Third Person Shooting* games.

The *remote user representation* represents the participating user or group of users at the other site. The aim of this representation form is to let this user or the group to appear present in the remote virtual environment. Therefore the factor of realism depends on the task of the users. Sometimes even more abstract user representations fit the requirements. Well-established methods of user representation are avatars and real time video textures. Research on avatars has produced from very abstract to very detailed human representation that include realistic visual and physical models [1]. Research on using real-time video is using stereoscopic or mono video and different texture mapping and image manipulation techniques [8]. The advantages of video conferencing are the high realism and the ease in handling of the video texture in order to position and scale it. The disadvantages are the transfer of video streams of the net and the match-moving of the texture with the virtual tool and input device representations selected by this user.

The *data model representation* is the data set of interest. Depending of the application these data sets can be a human body reconstructed from MR and CT recordings and a saw and drill for the surgeons, the car model with seats and crash test dummies for the engineers or the set of molecules for the chemistry professor. Data sets of interest can either be abstract models or reconstructed from scanner data for example. The best representation form is determined by the possibilities of scientific visualization and the user's task respectively. When interacting with the data the amount of possibilities which denote its functionality has to be represented (see also 2.6). Applications for experts exploit the real-world knowledge of the user which intuitively leads to the right way of interacting with the data whereas in virtual environments for training purposes functionality has to be represented in a perceivable way. There exist two main ways in VEs to show functionality to the user. One is to offer static menus which pack the whole set of operations that are applicable to the data sets. It is obvious that there are plenty of different possibilities to visualize these menus. When choosing this type of functionality representation the application designer and the programmer have to find the most suitable way to do this which is a really tough job. Problems which occur with those static menus are related to the limited interaction space of the displays systems and the uncomfortable usage when clicking through menu levels. It has been proven that it is a much better strategy to ask the data set what its functionality rather than to try to address a certain functionality with a selected tool. Then the data set's answer can be displayed as a menu again which is fixed positioned somewhere in the VE or attached to the user's gaze or hand[9, 7].

The *environment representation* reflects the ambience the users are working in. These representations can either be an operation theatre for surgeons, a lecture room for a professor and the students or a laboratory for a group of engineers. Environment representations are able to increase the feeling of immersion as the users feel more comfortable in their natural working environment than in an abstract one. Especially when using virtual environments for training purposes environment representations facilitate to transfer the learned in order to repeat it in real world.

The *virtual input device representations* reflect the active physical input device the user has chosen. These representations usually are virtual coloured rays when using the stylus or the multiple button devices. These rays enable the user to see where the physical input device or the hand points to. These representations facilitate the selection process.

The *virtual tool representations* reflect the active tool a user has chosen. These representations are 3D icons which are connected to the physical input device in use. Thus they follow the movements of the physical input devices or hands. With the help of these tool representations the user is aware of the possibilities of the active tools at any time.

2.5 The Application+Interaction Space

The Application+Interaction space describes how users interact, with each other and the data set of interest, collaboratively in the virtual environment. In order to find the best interaction we first have to understand the low-level makeup of interaction. Therefore we have to narrow down interaction tasks and to find interaction templates which are combinable to form more complex interactions.

2.5.1 Awareness-Action-Feedback Loops (AAF)

Awareness-Action-Feedback loops denote such interaction templates. These AAF loops give us the possibility to understand and analyse very tiny steps in interactions.

2.5.2 Autonomous AAF Loop

Before explaining complex collaborative interactions we start with autonomous interaction (see Figure 3).

The autonomous AAF loop is divided into four blocks. The first two blocks belong to the awareness phase where the user starts with proprioception as it was defined by Mine[9]. The proprioception lets the user be aware where s/he stands and looks to, the position and orientation of body parts like arms, hands and fingers and everything that is needed for interaction. This means that the user perceives itself in relation to the environment. The next step is to be aware of the physical input devices held in the users hands and the virtual tool representations connected

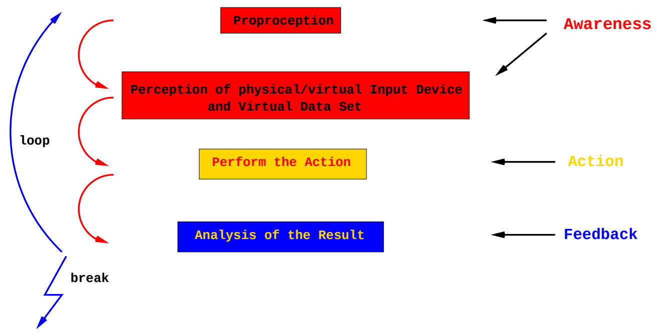

Figure 3: The Autonomous Awareness-Action-Feedback Loop.

to them. The position and orientation of the virtual data set is perceived in this phase as well. After the user is aware of the representation components and itself the action phase follows. This action can simply be to move the hand together with the physical input device. After the action phase the feedback phase follows. This feedback is meant to be action feedback without it would not be possible to analyse the result of the action. In this case the user perceives the movement of the virtual tool representations as s/he moved the input device together with the hand. After the perception of the status of the situation the user has to decide whether the task is completed and therefore wants to break the loop or whether the task is not completed yet and therefore prepares for the next action. We exemplify the AAF loop for the real scenario of a carpenter who wants to pound a nail into a piece of wood with a hammer. The steps of the AAF loop are:

1. **Proprioception** → *Awareness*
   Where am I ? Where do I look at ? Where are my hands, my fingers ?
2. **Perception of the physical/virtual input device and data set** → *Awareness*
   Where do I hold the stylus ? Is the hammer connected to my hand ? Where is the piece of wood ?
3. **Perform the action** → *Action*
   Interaction of human body (hands, fingers etc.) and physical input device. Position the nail on the wood and position the hammer !
4. **Result Analysis** → *Feedback*
   Perceiving the status of the situation. Perception of position, orientation and status of the virtual data and input device. (e.g. Did the data set allow to the operation ? Is the nail positioned

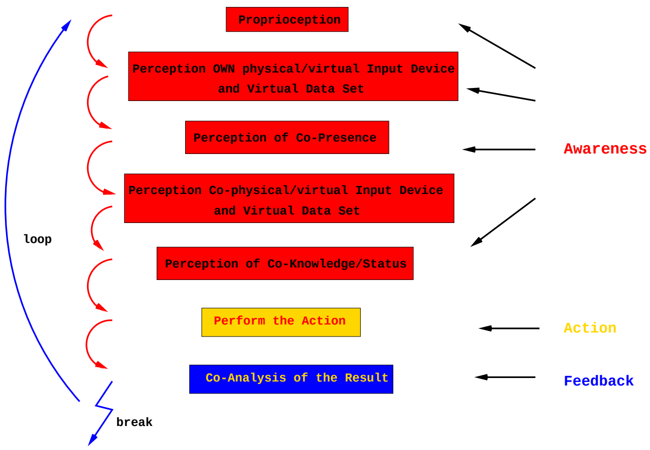

Figure 4: The Collaborative Awareness-Action-Feedback Loop.

correctly ? Is the hammer in place and ready to pound ?)
Depending on the status return to step 1. and proceed or break the loop (e.g. I am not ready yet so proceed with pounding the nail !)

5. Repetition of steps 1/2/3/4.

2.5.3 Collaborative AAF Loop

Collaborative Awareness-Action-Feedback loops are of the same structure as the autonomous AAF loops (see Figure 4).

The main difference between them is that the collaborative AAF loop has to address collaborative requirements that are necessary when working in a team. Again the collaborative AAF loop starts with the proprioception block and the perception of the own physical input devices and the virtual tool representations. After this but still in the awareness phase the user perceives the co-presence. It is comparable to proprioception but now information about the remote partner is queried like: Where is my partner, where does he look to, where are his hands, fingers etc.. Similar is the perception of the physical input device and the virtual tool representations together with the virtual data set. An interesting component represents the perception of co-knowledge and co-status. It is often not sufficient to know where you and your partner are located and where the object and the tools are when working in a team. We found out that knowing that your partner is aware of you is one of the most important steps in the awareness phase. To know that your partner is aware of what you are intending to do and how do you want to achieve this is essential for team work. Everything that supports this type of awareness increases the amount of collaboration. While perceiving the co-status the users check the situation. For the confirmation of this status check the users can do this by voice or with help of a gesture like "thumbs up". The action and the feedback phase abut to the already explained of the autonomous AAF loop. In order to apply the collaborative AAF loop to a real scenario we assume two carpenters who again want to pound a nail into a piece of wood with a hammer. One carpenter holds and positions the nail on a piece of wood and the other carpenter pounds the nail with a hammer. We are then able to describe the whole interaction task from the sight of the carpenter who holds the hammer like done in the following.

1. **Proprioception** → *Awareness*
The same as before (see AAF loop).

2. **Perception of the physical/virtual input device and data set** → *Awareness*
The same as before (see AAF loop).

3. **Perception of co-presence** → *Awareness*
Where is my partner ? Where are his hands and fingers ? Where does he look to ?

4. **Perception of co-physical/co-virtual input device and data set** → *Awareness*
Where does my partner hold the nail and the wood ? How is the relationship between nail and wood ?

5. **Perception of co-knowledge and co-status** → *Awareness*
Is my partner aware of me ? Does he know where I am, where I am looking to and where I hold the hammer ? Does he know what I am doing and what I want to do ? Is everything ready now ? Confirmation of the status check by voice or "thumbs up".

6. **Perform the action** → *Action*
The same as before (see AAF loop).

7. **Result Analysis** → *Feedback*
The same as before (see AAF loop).

8. The steps 1. to 7. are repeated until the task is finished.

2.6 Operations, Metaphors, Interaction Techniques

Awareness-Action-Feedback loops like shown in the Figures 3 and 4 are templates. With the help of operations, metaphors and interaction techniques it is now possible to give those templates a "face". This means that depending on the user's subtask the appropriate operations, metaphors and interaction techniques have to be chosen for each action.
Operations defined in our taxonomy provide the means for supporting manipulation of virtual data and shared

manipulation between remote participants. They describe what can be done with the virtual data in terms of how the data can be explored. They can be data independent (i.e. basic operations such as selecting), or data dependent (i.e. slice through a 3D volume of data).
Metaphors for interaction and collaboration make use of everyday interaction and collaboration paradigms to provide intuitive ways of interaction in virtual environments (i.e the metaphor of working around a table). We distinguish between three different kinds of metaphors. *Stand-alone Metaphors* such as walk, fly and teleport, directly use or extent real-life paradigms to allow navigation through a virtual environment. *Content specific metaphors* that allow the user to focus on the part of data set of interest, look closer, hear/touch interesting subpart, as well as additional ones like play video/TV, search information library, can also be adapted from real-life paradigms. *Collaborative Metaphors* are visual and verbal communication between users and sharing viewpoints of participants.
Finally, *interaction techniques*, complement the metaphors by determining how to support and implement the different types of operations[7].

3 Application Design

To design a collaborative virtual environment that supports the above requirements, we carefully studied all the issues mentioned in earlier sections of this paper, in order to select the most appropriate representation components, metaphors, operations and interaction techniques. The task description is as follows:
Two users, a medical professor and a medical student work together on a virtual human data set. They stand opposite each other around a table. They are able to walk around the table and to have a look from the other side onto the data. The data set consists of a human skin and an underlying skeleton and heart model. Both users are able to cut the skin in order to see the underlying bones and inner organs, to pick bones and to drag them. The data set is used for anatomical education. Names of all bones can be queried, test scenarios, where a set of bones has to be inserted into the skeleton, can be uploaded. The two users are equal in their possibilities to work on the data set.
After the UTA and the definition of the UNS we came up with the following application design (see Figure 5).
Generic operations such as selecting, zooming, translating, pushing, dragging, grabbing, highlighting and content specific ones, such as labelling of parts of the data sets, cutting, slicing planes, starting/ending video conferencing, were included in the design of the system. We decided to use menus and virtual pick-rays as interaction technique to apply the desired operations to the data sets. Therefore the generic operations are applied using a fixed toolbar with a rotate tool, translate tool, zoom tool, drag and push tool. The content specific operations allow slicing of the 3D representation of the patient's data. These operations are applied by calling an Object bound ring menu. The toolbar is fixed whereas the ring menu, bound to the object, disappears when an operation has been selected. Additional content specific operations for the real patient data sets are colour lookup sliders, compass to obtain the orientation when slipping into the data set, slicing and clipping planes. For the skeleton model content specific operations for material change and fade, and wire-frame and gray value windows are available. Additional operations include viewing of labels bound to different bones, or of animation of the virtual heart model. Additional visualization of interesting medical information is at the user's disposal. Rendering of video sequences in mono or stereo on virtual Screens is also part of the system, to allow video sequences of endoscopic recordings of the stomach or the esophagus to be played at will. As interaction devices in our prototype we use tracked Crystal Eyes shutter glasses, a Polhemus stylus, and for two handed interaction, a three button tool also tracked by the Polhemus Fastrak system. In order to enable teamwork we implemented the following metaphors:

- ring up the remote partner
- join a remote session
- share a tool
- face-to-face communication
- tug of war

The ring-up and join session metaphors were implemented by providing a session name. As soon as the user connects to a session a whole copy of the virtual scene provided by the others is transferred to the local site. In the same moment a video/audio connection to the other Responsive Workbench is established (see Figure 5). The video screen with the remote partner provides the content specific operation to mute or disconnect this video/audio conferencing depending on user's wish. To enable collaborative manipulation of the data, the generic toolbar is distributed together with the patient's body and skeleton model. The content specific operations are also shared since there are bound to the shared data sets. The metaphors we make use of in the collaborative case are the face-to-face communication and mirrored viewpoint or sharing viewpoint (look through other's eyes and/or look over other's shoulder). Finally for the collaborative manipulation we used the tug of war metaphor (see Figure 5).

3.1 Technical Details

For rendering two SGI ONYX IR2 workstations are used with two graphics pipes and six R12000 processors each. Electromagnetic Fastrack tracking systems from Polhemus are used to track the head and the two input devices, a Polhemus stylus and an own built three button tool. For communication purposes wireless microphones and headphones are available. The video and the audio conferencing is handled by two O2 workstations. Video streams in PAL resolution are grabbed directly from the infra-red video camera, compression using motion jpeg compression and sent over the fast ethernet network to another O2 workstation. There the stream is decompressed and fed into the DIVO boards of the ONYX. The same O2 which handles the video conferencing manages the audio conferencing. The audio stream grabbed from the wireless microphones is compressed and then send to the other O2 where the headphones are plugged in. The software framework we are using is AVANGO developed by GMD. It combines the familiar programming model of existing stand-alone toolkits with built-in support for data distribution that is almost transparent to the application developer. A detailed description of the toolkit and the way distribution is implemented can be found in [10]. A schematic of the built setup is shown in Figure 6.

Figure 5: Image taken from a current session. In order to take an image of the session we had to render this scene in mono.

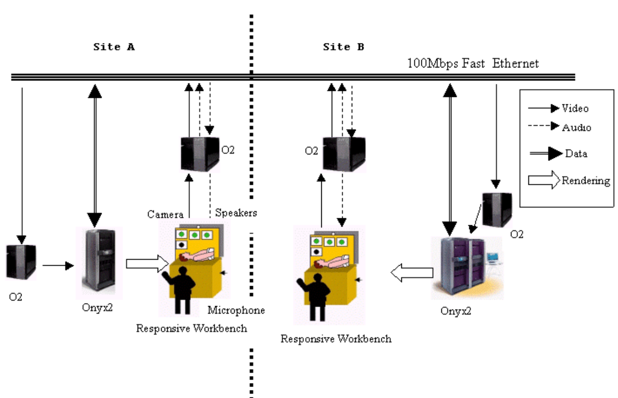

Figure 6: Schematic of the used setup.

4 Conclusions and Future Work

We presented our vision in creating Collaborative Virtual Environments that provide distributed collaborative teams with a virtual space where they could meet as if face-to-face, coexist and collaborate while sharing and manipulating in real time the set of virtual data of interest. We discussed the issues involved in bringing together Human Computer Interaction and Human to Human Communication, focusing on projection-based Virtual Environment systems.

The initial evaluation of the prototype was based on heuristic analysis [5] and we are planning to extend it to detailed user-task and ergonomic analysis [5].

References

[1] T. Capin, I. Pandzic, N. Magnenat-Thalmann, and D. Thalmann. Avatars in networked virtual environments. *John Wiley and Sons*, 1999.

[2] C.A. Ellis, S.J. Gibbs, and G.L. Rein. Groupware - some issues and experiences. *Communications of the ACM*, 34(1):38–58, 1991.

[3] I. Foster, J. Insley, C. Kesselman, and M. Thiebaux. Distance visualization: Data exploration on the grid. *IEEE Computer*, 32(12):36–43, December 1999.

[4] B. Froehlich, S. Barrass, B. Zehner, J. Plate, and M. Goebel. Exploring GeoScience Data in Virtual Environments. In *Proc. Visualization 99 (to be published)*, 1999.

[5] J.L. Gabbard, D. Hix, and J.E. Swan II. User-centered design and evaluation of virtual environments. *IEEE Computer Graphics and Applications*, 19(6):51–59, November/December 1999.

[6] G. Goebbels, W. Frings, T. Eickermann, F. Hossfeld, and S. Posse. Global broadcast - supercomputer-enhanced functional mri of the human brain. *In IEEE Concurrency*, 8(1):11–13, Jan.-Mar. 2000.

[7] G. Goebbels, V. Lalioti, and M. Goebel. On collaboration in distributed virtual environments. *In Proceedings of HC-2000 - Third International Conference on Human and Computer*, Sep. 2000.

[8] V. Lalioti, F. Hasenbrink, and C. Garcia. Meet.me@cyberstage: towards immersive telepresence. *Virtual Environments'98, 4th Eurographics Workshop, Germany*, 16-18 June 1998.

[9] M. Mine, P. Frederick, Jr. Brooks, and C. Sequin. Moving objects in space: Exploiting proprioception in virtual-environment interaction. *Proceedings of SIGGRAPH 97, Los Angeles, CA*, 1997.

[10] H. Tramberend. Avocado: A Distributed Virtual Reality Framework. In *Proc. of the IEEE Virtual Reality*, 1999.