Team work in Distributed Collaborative Virtual Environments

by

Gernot Peter Josef Goebbels

Submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy in Computer Science

in the Faculty of Natural & Agricultural Sciences

University of Pretoria

Pretoria

November 2001

Team work in Distributed Collaborative Virtual Environments

by

Gernot Goebbels

Today's technology and advances in networking and telecommunications stimulate a change in the way everyday business is carried out, making it a globally distributed process, in which communication and collaboration of geographically dispersed groups is of vital importance. Virtual Environments are adapting accordingly, by providing not only a better man-machine interface, but also by facilitating human-to-human interaction and collaboration over distance.

Therefore, new challenges are introduced in terms of distribution and interaction in Virtual Environments. It is not only a question of solving the technical problems of gathering and transmitting multimedia data streams with sufficient quality and speed, but also a question of addressing the specific needs of human communication and collaboration.

The vision of Collaborative Virtual Environments (CVE) is to provide distributed, collaborative teams with a virtual space where they can meet as if face-to-face, co-exist and collaborate while sharing and manipulating in real-time the virtual data of interest.

The objective of this thesis is to provide the Virtual Environments research community with a thorough investigation of distributed, collaborative interaction between geographically dispersed teams using projection based Collaborative Virtual Environments.

Thesis Supervisor: Prof. Dr. Vali Lalioti
Department of Computer Science

Submitted in partial fulfillment of the requirements for the degree Doctor of Philosophy in Computer Science

Acknowledgements

While this thesis is my own work, many people to whom I wish to express my most sincere thanks supported it.

Firstly, I would like to acknowledge my first supervisor Prof.Dr. Vali Lalioti for her competent comments and her never ending support and encouragement of my work. Further, the members of the PhD committee and co-supervisors as well as the head of the Computer Science Department Prof.Dr. Derrick G. Kourie.

In addition, I would like to thank the staff at GMD's Department of Virtual Environments for the best collegial and co-operative atmosphere I ever found all over the world among the places I have ever been working at. I am particularly grateful for the way Dr. Martin Göbel is heading this group and the way he provides us with an amazingly creative and productive working environment.
Additionally, I would like to thank all the evaluators who volunteered their time assessing the system.

Finally I would like to thank my family. My parents Josef and Karen Goebbels and my beloved grandmother Gertrud Goebbels who supported me throughout my whole life providing me with the best education possible. Especially I would like to thank Yvonne Reiter who has always endured my long working hours uncomplaining, while remaining true and faithful and with a good sense of humor, supportive and enthusiastic towards everything I do.

Contents

List of Figures

Introduction

The motivation for providing multi-sensorial interfaces for human-computer interaction is rooted in the nature of human perception and cognition, which uses several sensory channels at a time to construct what is generally referred to as reality.

Naturally, the more sensory channels are stimulated coherently in a man-machine interface, the richer the interaction models. The more of our innate and culturally acquired perceptual and cognitive skills are exploited in an interface, the more refined and efficient the interaction may be. This is especially valid for interfaces which mimic to a large extent certain aspects of our everyday environment to create what we call Virtual Environments (VE).

One of the underlying assumptions of these interfaces is that the more a Virtual Environment perceptually resembles the environment we are familiar with, the easier it will be for us to orient, navigate, and act in such an environment. But it has to be seriously doubted if technology will ever be able to create a synthetic sensory experience completely indistinguishable from the one we experience in our everyday world. Fortunately this is not a drawback of Virtual Reality technology but its most interesting aspect as it forces the designers and developers to create efficient interaction techniques and metaphors which refer to our cognitive skills but which do not necessarily attempt to mimic interaction as it happens in everyday life. This is how new interaction techniques evolve and become candidates for developing a new framework or language of expression. Virtual Environments are currently developing their own language of expression which is still very rudimentary.

Therefore, **chapter 1** provides a detailed survey on Virtual Environments. This survey includes various display systems and input devices for interaction. In addition, selected software toolkits are introduced. The particular reason for choosing these software toolkits is their uniqueness with respect to their special capabilities for implementing Virtual Environments.

Today's technology and advances in networking and telecommunications stimulate a change in the way everyday business is carried out, making it a globally distributed process, in which communication and collaboration of geographically dispersed groups is of vital importance.

Virtual Environments are adapting accordingly, by providing not only a better man-machine interface, but also by facilitating human-to-human interaction and collaboration over distance.
Therefore, new challenges are introduced in terms of distribution and interaction in Virtual Environments. It is not only a question of solving the technical problems of gathering and transmitting multimedia data streams with sufficient quality and speed, but also a question of addressing the specific needs of human communication and collaboration.

The vision of Collaborative Virtual Environments (CVE) is to provide distributed, collaborative teams with a virtual space where they can meet as if face-to-face, co-exist and collaborate while sharing and manipulating in real-time the virtual data of interest.
The objective of this thesis is to provide the Virtual Environments research community with a thorough investigation of distributed, collaborative interaction between geographically dispersed teams using projection based Collaborative Virtual Environments.

The need for such high-end Collaborative Virtual Environments is becoming more pressing due to the globalized nature of today's market. Distributed companies are more common and their business requires not only a distributed structure but also effective collaboration over distance in order to minimize time and travel costs. Another category of businesses where there is a need for Collaborative Virtual Environments that integrate tele-conferencing facilities is businesses where the raw data are gathered from remote areas while the experts and the high-end infrastructure for visualization are located on the company's sites.

Collaborative Virtual Environments that include face-to-face communication can also greatly benefit remote consultation and tele-education, providing means of accessing the experts or infrastructure that are not available at the consultation or education site.

The advances in networking and high performance computing can provide the basis for such advanced Collaborative Virtual Environments. However, this is not enough. Since Virtual Environments denote interactive computer simulated worlds, the implemented interaction techniques and visual representation components have the highest impact on the usability and thus on the acceptance of the designed CVE.

Therefore, **chapter 2** deals with a detailed comparison of interaction techniques. It is reviewed whether these interaction techniques can be used to complete the five basic interaction tasks, selection, position, text and numeric input as well as confirmation [39]. It is also reviewed whether these techniques are usable in Virtual Environments [1]. Additionally all related work relevant to this thesis is presented. This includes additional new three-dimensional interaction techniques and a survey on selected Collaborative Virtual Environments. The particular reason for choosing these Collaborative Virtual Environment systems is the fact that they specifically address the needs of users working cooperatively together in a Virtual Environment. Teleport and the NTII were chosen due to their use of tele-immersion approaches. The objectives of the applications are introduced and it is shown how interaction in these Collaborative Virtual Environments is designed.

In general interaction can be seen as cultural techniques of expression that tend to mix and merge, reference each other, and are transformed and rethought in the context of new media. They form the rich tissue of a culture's means of expression most consequently explored and developed in its art. It is not necessary to get into contemporary art theory in order to illustrate what this means. An analogy is evident in today's advertisement design which more and more refers to the desktop metaphor of current computer graphics user interfaces. The concept of a window, e menu-bar or pull-down menus suddenly can be used to present different aspects of a product. Such an advertisement never would have been understood before a significant fraction of members of a society became acquainted with modern human-computer interfaces.

However, for the Virtual Environment interaction designer it is a very complicated and challenging task to create a user interface that fits the user's requirements. Problems with the design of user interfaces are manifested in the lack of user interface standards for Virtual and Collaborative Virtual Environments. Furthermore there are no formalizing approaches which are capable of supporting the implementation of Collaborative Virtual Environments with design guidelines.

In order to overcome this problem the principle aim of **chapter 3** is the development of a theoretical interaction approach for Virtual and Collaborative

Virtual Environments.
The approach includes a taxonomy that creates, from a varied array of VE/CVE influence components, a hierarchy of groupings that have an orderly relationship to each other. This taxonomy is usable for the categorization of hardware and visual representation components supporting the user's awareness in the Virtual Environment. Further it categorizes operations, interaction metaphors and interaction techniques which mainly influence the user's interactions with the environment.
The developed taxonomy is not a hierarchy of classified VEs but a hierarchy of classified influence factors that show an impact on the design process of VEs and CVEs. With the help of this taxonomy of influence factors it is then possible to develop a VE/CVE design model which supports the VE/CVE designer to consider the large amount of these influence factors. Thereby the design model also shows the dependency of the influence factors and enables to simulate the appearance of the VE/CVE early in the design process.
The input for this simulation delivers the requirement engineering process that uses the task description and task analysis which determine the User+Need Space. An example for such a simulation is given at the end of chapter 3.
The reason for the development of a new VE/CVE design model instead of taking an existing one is that the existing models developed in the CSCW and HCI community do not pay enough attention at Human-to-Human communication and collaboration in large scale projection based Virtual Environments. Bowman et.al., for example, developed a taxonomy of different techniques concerning the three tasks navigation/locomotion, selection and manipulation. This taxonomy enables the VE designer to find the interaction technique best-suited for a given task. The orderly relationship between the classified techniques is obtained according to usability issues taking into account user input devices and tasks.
Bowman's taxonomy has been developed for and evaluated in HMD based systems (Head Mounted Displays) but not in projection based displays. Whether this developed taxonomy is valid also for projection based display systems needs to be evaluated in future.

In order to find the best interaction within this thesis the low-level makeup of interaction is analysed. Interaction tasks are narrowed down and interaction templates are defined which can be combined to form more complex interactions. The developed *Awareness-Action-Feedback* loops denote such interaction templates. These loops provide the possibility to understand and analyze very tiny steps in interactions. With their help it is possible to track down usability problems early in the design phase.
The taxonomy presented in this thesis attempts to group and categorize all the

influence factors. It is a practical tool that can be used as a framework for the design and evaluation of CVEs as shown in this thesis. Therefore, the utility of the taxonomy is considered rather than its absoluteness or completeness. The objective is to facilitate guided design of applications for supporting team work in CVEs.
In the last section of chapter 3 the simulation of an example CVE design is performed. With this example the reader will be able to understand the make-up and the process of CVE design making use of the VE/CVE design model developed from the taxonomy of influence factors at the beginning of this chapter.

Chapter 4 guides the reader through the whole design process of an Collaborative Virtual Environment. In this context it is illustrated how all the components of the theoretical interaction taxonomy can be put into practise. This also includes a detailed task description and analysis of the user requirements.

A technical description of the implementation of two Collaborative Virtual Environments follows in **chapter 5**. Firstly, the setup is described with respect to the used hardware configuration introducing the rendering and interaction equipment. The remainder of this chapter presents the software configuration describing the audio/video conferencing as well as rendering and distribution using the Avango software framework. Code fragments represent parts of the application programming and flow charts illustrate the combination of different techniques and equipment.

However, when implementing Virtual and Collaborative Virtual Environments designers and programmers usually tend to guess about the best realization and implementation of interaction techniques or even the whole application [53]. Many works have shown that user based assessment is an essential component of developing interactive applications and in this work it is shown that user based assessment is especially important for applications as complex and innovative as CVEs. Already the assessment of parts of the application by different users except the designers can substantiate or refute realizations of a specific Collaborative Virtual Environment. If those assessments are formalized they are called evaluations. But still there is a lack of formal approaches for efficiently carrying out evaluation of Collaborative Virtual Environments.

These evaluations are crucial for the implementation of Collaborative Virtual Environments because errors made in the early phases of the design are the mostly costly to repair later on. It is also a very delicate activity, as it requires heavy user involvements and evaluation approaches that are capable of assessing usability aspects. There exist four different approaches [79]. These are the:

- Interaction oriented approach
- User oriented approach
- Product oriented approach
- Formal approach

The *interaction oriented approach* is the most common one and is concerned with all kinds of usability testing with users. The *user oriented approach* tries to measure usability quality in terms of mental effort and attitude of the users. This is done by using questionnaires and interviews. The *product oriented approach* is concerned with measuring ergonomic attributes and thus quantitative measurements are necessary. Lastly the *formal approach* tries to simulate usability in terms of formal models. This approach can be seen in the context of theory based evaluation.

Therefore, **chapter 6** deals with the evaluation of the implemented applications according to usability and collaborative awareness. For performing intelligent evaluation, specific questionnaires and evaluation items are designed as the focus in this thesis is the user oriented approach.
For the statistical analysis of the numeric evaluation results the average values and the corresponding expectancy values are computed. However, for obtaining more subtle results which allow for the formulation of CVE implementation guidelines supporting team work a new analysis method is developed, namely the Variation Group Analysis.

In **chapter 7** a review of the results of this thesis is presented. Additionally to the developed interaction taxonomy for Collaborative Virtual Environments the defined Awareness-Action-Feedback loops are concluded. Furthermore the results of the evaluation assessing usability and collaborative awareness are listed in detail.
The chapter concludes with a discussion on possible enhancements and directions for future research. These are listed in detail trying to encourage further work in this area making use of the results of this thesis as a basic approach.

Appendix A is a glossary on special terms that are used in Virtual Environment technology. It might help to understand different expressions that are not used in the everyday language. In addition, it can be used by non-expert evaluators to assess the different aspects of usability since they are not familiar with the topic.

Appendix B is an example of a code implementation using the scripting interface of Avango. This scheme code creates the necessary scene graph for stereo video conferencing.

The **appendices C to G** present the designed questionnaires used for the usability evaluation. Appendix C is querying general information about the user in order to create a user profile. The remaining appendices are used for assessments of the evaluation items that are worked out in this thesis.

This thesis is entirely developed at the Department of Virtual Environments of the German National Research Center for Information Technology (GMD) in Sankt Augustin, Germany.

Chapter 1

Virtual Environments

In this chapter a detailed survey on the current technology in Virtual Environments (VE) that allows the highest degree of intuitive interaction is given. This includes various display systems and input devices as well as most important software toolkits.

1.1 Display Systems

According to [44] there exist three different classes of display systems. CRT-based displays such as Head-Mounted Displays, Virtual Model Displays such as Responsive Workbenches and ReachINs, and Spatially Immersive Displays (SID) such as CAVEs and DOMEs. It is unreasonable to assume that a single display device could support any or all VE tasks equally well. Instead it is obvious from the proliferation of display types that most display types are perfectly suited for some task, sufficient for some other tasks, and ill-suited, impossible, or intractable for others. Thus, to assess the effectiveness of a display device is to assess how well the device matches representative user tasks.

1.1.1 Head Mounted Displays (HMD)

In its simplest form, an HMD consists of an image source and collimating optics in a head mount [94]. The HMD can then become more elaborate in several ways. There are one or two display channels. These channels display graphics and symbology with or without video overlay. They can be viewed directly and occlude external vision for a fully immersive experience, or a semitransparent combiner is used with see-through capabilities to the outside world. In this augmented reality mode, the HMD is able to overlay symbology, or other information on the world view. The HMD image source can either be CRT

(Cathode Ray Tube (small TV-tube)), LCD (Liquid Crystal Display) or in the near future RSD (Retinal Scanning display) mounted on the head, or the image is brought up to the head through a fiber-optic bundle. An HMD uses a simple headband for mounting on the head, or the optics and the displays are integrated into an aviator's flight helmet. This latter device is a specialized case of the HMD, called the helmet mounted display. HMDs have often been used to display Virtual Environments in the past. Nowadays they are not used very often due to their uncomfortable usage, the level of contamination of the CRT close to the eyes and the low resolution. However, in Augmented Reality see-through HMDs are still used quite often.
Typical application areas come from the manufacturing industries where real objects are overlayed with virtual information providing, for example, assembling aid.

1.1.2 ReachIn Display

The ReachIn display is a unique Human-Computer Interface (see Figure 1.1). The system combines a stereo-visual display, a haptic device and a six degree of freedom positioner. The innovative use of a semi-transparent mirror creates an interface where graphics and haptics are co-located. The user interacts with the virtual world using one hand for navigation and control and the other hand to touch and feel the virtual objects. It is possible to see and feel the object in the same place.
The description of an excellent interface design for the ReachIn display system

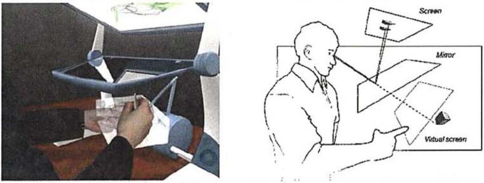


Figure 1.1: The ReachIn is a very intuitive display system for many kinds of applications. [ReachIn Inc.,2001]

can be found in [76]. ReachIn displays are typically used for medical training, product development and design engineering.

1.1.3 Responsive Workbench

The Responsive Workbench in Figure 1.2 is a high resolution tabletop display system which has been developed at the Department of Virtual Environments of the *German National Research Center for Information Technology (GMD)* [63, 64]. The Responsive Workbench consists of a RGB projector

Figure 1.2: The Responsive Workbench provides a natural metaphor for visualizing and interacting with three-dimensional computer-generated scenes. On the right hand side two engineers are working collaboratively in front of the same display.

which projects stereo images over a mirror onto a display plane. This projection plane is sloped by 20°. A separate image is computed for each eye, and the computer quickly alternates the display of the two views. Users wear shutter glasses, which cover the left eye while the right eyes image is displayed, and vice versa, thus producing the stereoscopic effect. A Polhemus six degrees of freedom sensor is attached to the shutter glasses for head tracking. This allows the system to compute the correct perspective image for any user location. The users interact directly with three dimensional virtual objects within the viewing frustum using six degrees of freedom input devices.
Typical application areas of the Responsive Workbench are delivered by engineering and medical visualization.

1.1.4 Two-sided Responsive Workbench

The new two-sided Responsive Workbench has a horizontal and a vertical display, which are smoothly adjacent (Figure 1.3). By extending the normal

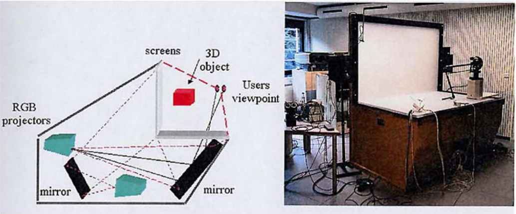


Figure 1.3: The two-sided Responsive Workbench is extended by an additional vertical screen. Thus the viewing frustum is substantially increased and virtual objects can be observed at the user's eye level. On the right hand side a Phantom Force Feedback device is integrated into the RWB's interaction space.

Responsive Workbench with an additional vertical screen, the viewing frustum is substantially increased and virtual objects can be observed at the user's eye level which was not possible before. Using current PC and workstation technology the two-sided Responsive Workbench is operated with a resolution of $1280 * 1024 * 96Hz$ stereo. Because of the extended viewing frustum the two-sided Responsive Workbench can be used in more application areas than the one-sided Responsive Workbench.

The two-sided Responsive Workbench can be used for engineering and medical visualization like the one-sided Responsive Workbench. Product design as well as various training applications are also applicable on the two-sided RWB.

It is important to mention that throughout this thesis the display system term RWB is refereing to the two-sided Responsive Workbench instead of the one-sided older version.

1.1.5 CAVE - CyberStage

The CyberStage, an improved version of the CAVE [27] has been developed at the Department of Virtual Environments of the *German National Research Center for Information Technology (GMD)*. The immersive features of the

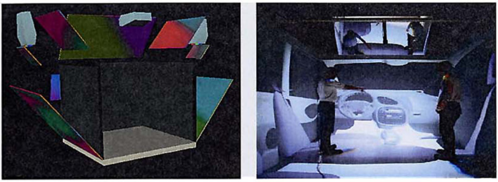


Figure 1.4: The CyberStage uses three mirrors for the side projections and either one mirror for the floor and front projection. Four sub-woofers are built into the floor allowing for rendering low frequency signals perceivable through feet and legs. On the right hand side two user's are working on the Volkswagen Sharan interior design.

CyberStage are based on four-side stereo image projection and eight-channel spatial sound projection both controlled by the position of the user's head followed by a tracking system. The sound projection is complemented by vibration emitters built into the floor of the system and allow for rendering low frequency signals perceivable through feet and legs. Therefore the CyberStage can create the illusion of presence in virtual spaces using various interfaces and interaction metaphors which visually and acoustically respond to the users' actions. These interfaces allow for navigation in virtual spaces and manipulation of virtual objects.
The design of the CyberStage was aiming at providing a high degree of immersion and presence for a wide spectrum of Virtual Environments and it has already been used in the following application areas:

- scientific visualization/sonification
- industrial data exploration
- product presentation and marketing
- product design and evaluation
- architectural planning and design
- exhibition design and evaluation
- media art installations and promotional infotainment

1.1.6 Powerwall

The Powerwall is a huge multi channel rear projection screen. It can have dimensions up to 8m wide and 3m high. The Powerwall consists of mostly two or three segments, where each segment is projected by one or two graphic channels, which in turn are powered by one or two projectors. Therefore in addition to three channel mono displays and six channel stereo displays it is also possible to have a six channel mono display. By using double projection for each segment, strong light-intensity and high brilliance is achieved. Light reflections and surface dispersions can be analyzed as good as with real finished models.
Powerwalls are typically used in automotive industry. Especially product designers have a suitable capability to visualize a complete car in a 1:1 scale.

1.1.7 Cylindrical Displays

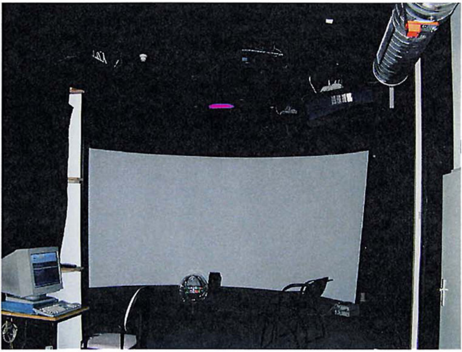


Figure 1.5: GMD's cylindrical display system is a 230° projection with 7m diameter.

Cylindrical displays are multi channel panoramic projections on cylindrical or spherical curved surfaces with a field of view up to 360°. Similar to the Powerwall, the Cylinder consists of multiple (mostly 3 for 180°) segments,

where each segment is projected by one channel powered by one projector. *Soft-edge-blending* creates a seamless and invisible integration of the separate images to form a unique full screen image of a very high quality. Cylindrical displays can be driven with analogue CRT projectors or with digital DLP projectors.
Cylindrical projection displays are ideally suited for a variety of applications in the presentation and simulation area due to their capabilities to host groups of up to 10 users.

1.2 Input Devices

There exist many input devices as many graphics applications require the input of three dimensional co-ordinates in order to position objects in a virtual space. These devices can be classified into devices for the hands and for the feet. Devices which are used by the hands are: keyboard, mouse, joystick, trackball, stylus, touch screen, digitizer (graphics tablet), data and pinch glove, space mouse, cubic mouse, multiple button tools etc.
Input devices used with the feet are: Treadmill (for navigation simulation in Virtual Environments), driving wheels including pedals (for flight and racing games or simulators).
Not belonging to one of these two groups microphones and video cameras are also considered to be input devices.
Desktop applications typically use a mouse, trackball or joysticks for such input. Applications developed for Head-mounted displays mostly use gloves such as pinch gloves or data gloves. A variety of input devices are used or developed especially for projection based display systems. In this section a survey of the most important input devices is provided including multi-sensory feedback.

1.2.1 Pen-like Input Devices

In order to interact within a Virtual Environment, devices are needed to track the location of real objects. As this world is three dimensional, three measures should be used (X,Y and Z). The mouse and the joystick are examples of such tracked control devices. However, in Virtual Environments it is not sufficient to know the position of the input device only, the orientation is needed too. Sensors which measure position and orientation are usually electromagnetical or optical devices. But also ultra-sound based systems and inertial trackers are available.

Figure 1.6: The left image shows a pen-like stylus input device from Polhemus. On the right hand side pinch gloves are shown.

1.2.2 Pinch Gloves

Another common control device is the pinch glove (see Figure 1.6). This pair of stretch-fabric gloves contains sensors in each fingertip which detect contact between the digits of the user's hand. These gestures can be used for a wide range of control and interactive functions. Any combination of single or multiple contacts between two or more digits can be programmed to have specific meanings, ranging from simple on/off to multi-part, multi-action commands. Thus the pinch glove system provides a reliable and low-cost method of recognizing natural gestures. Recognizable gestures have natural meaning to the user: a pinching gesture can be used to grab a virtual object, and a finger snap between the middle finger and thumb can be used to initiate an action.
As it is very easy to integrate, the pinch glove system can be integrated into driving and flight simulators, interactive 3D video games or any application that requires a wide range of tactile gestures.

1.2.3 Cubic Mouse

The Cubic Mouse is a novel and intuitive input device which has been developed at the Department of Virtual Environments of the *German National Research Center for Information Technology (GMD)* [41, 42]. This device consists of a cube-shaped case, three rods, and controls buttons (see Figure 1.7). Each rod passes through the center of two parallel faces of the case. The rods are perpendicular to each other and movable. They represent the X, Y, and Z

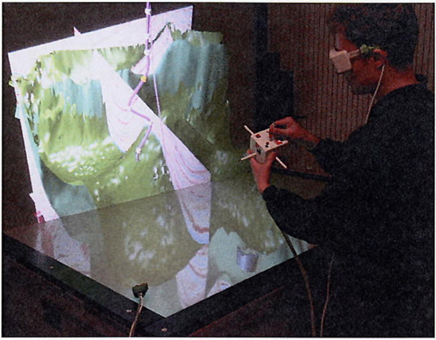


Figure 1.7: The user controls a geo-scientific scenario at the Responsive Workbench using the Cubic Mouse device.

axes of a co-ordinate system. In addition to the rods a six degree of freedom tracker sensor is embedded in the cube-shaped case. This tracker sensor is used to orient and position the virtual world in three dimensional space relative to the observer. In this way the rods stay aligned with the co-ordinate system axes. By pushing and pulling the rods motions of virtual objects are specified constrained along the X, Y, and Z axes. Typically the users hold the device in their non-dominant hand to position and orient the world, while the dominant hand operates the rods and the control buttons.
The Cubic Mouse is typically used for controlling volumetric data sets in geo-science and automotive applications. Selection tasks are rather complicated to perform.

1.2.4 Phantom Force Feedback Device

Force feedback provides direct perception of three-dimensional objects and directly couples input and output between the computer and user. It acts as a powerful addition to graphics display for problems that involve understanding of 3D structure, shape perception, and force fields [22]. There exist many force and tactile feedback devices. A well known one is the Phantom Force Feedback device from Sensable Inc. (see Figure 1.3 and Figure 1.8). The Phantom

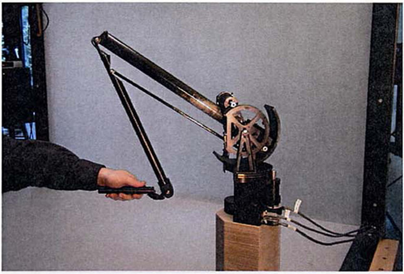


Figure 1.8: The Phantom Force Feedback Device integrated into the Responsive Workbench's interaction space. [Sensable Inc.,2001]

six degree of freedom prototype is a desk mounted force feedback system that provides six degree of freedom force and torque feedback. The system consists of the device itself, accompanying power electronics, a remote/safety switch, and a PCI interface card. The device is leveraged from haptic technology developed and implemented in standard Phantom three degree of freedom force feedback devices.
The addition of force display to computer graphic systems has produced a factor of two improvement in rigid-body motions during simulated drug docking, enabled new types of manipulation, and provided users with a stronger sense of understanding [82]. For surgery training applications tactile and force feedback is of essential importance.

1.3 Virtual Environment Software Toolkits

Many toolkits for the development of stand-alone VE applications exist today. They provide the programmer with a high-level interface to represent complex geometry in a scene graph and to render that scene graph. The programmer is shielded from the details of dealing with low-level graphics and system APIs, and can concentrate on the development of the application itself. Several attempts to offer similar toolkits for distributed VE application development have been made in recent years (eg. DIVE [24], VR Juggler [5], WTK [84, 85]). This section provides a survey on some important VE software toolkits. The particular reason for choosing these software toolkits is their uniqueness with

respect to their special capabilities for implementing Virtual Environments.
VR Juggler and Avango are unique in terms of their real-time rendering capabilities. Also the easy configuration for different display systems and input devices make them being different compared to other software toolkits such as e.g. Massive etc.
Also the scripting interface that allows for run-time modifications and easy prototyping is a outstanding feature.
DIVE on the other hand is a VR software toolkit that enables users to share and manipulate data very effectively and further allows different types of communication.
The following sections are presenting the mentioned toolkits in more detail.

1.3.1 DIVE

The Distributed Interactive Virtual Environment (DIVE) is an internet-based multi-user VR system where participants navigate in three dimensional space and see, meet and interact with other users and applications [24]. The key requirements for a Virtual Environment development system that the DIVE's design addresses are: performance, extensibility, and portability.
The first DIVE version appeared in 1991. The DIVE software is a research prototype covered by licenses. Binaries for non-commercial use, however, are freely available for a number of platforms. The software supports the development of Virtual Environments, user interfaces and shared synthetic applications. DIVE is especially tuned to multi-user applications, where several networked participants interact over a network. Dynamic behaviour of objects are described by interpretative Tcl scripts evaluated on any node where the object is replicated. Scripts are triggered by events in the system, such as user interaction signals, timers, collisions, etc. DIVE imports and exports VRML and several other formats. It is integrated with the World-Wide-Web and is HTTP/FTP/HTML/MIME compliant.
DIVE applications and activities include virtual battlefields, spatial models of interaction, virtual agents, real-world robot control and multi-modal interaction [40]. Through its various versions, DIVE is available on the almost all platforms.

1.3.2 VR Juggler

VR Juggler is an open source virtual platform for virtual reality application development created at the Virtual Reality Applications Center at Iowa State University [5] and is the follow-up of the CaveLib project, although the new architecture is very different.

Applications developed in VR Juggler can transparently move between a wide range of VR systems, operating systems, and graphics APIs. VR juggler's design addresses several key requirements for a Virtual Environment development system namely, performance, flexibility, rapid application development, extensibility, and portability.
Flexibility is achieved by placing common abstractions over I/O devices. New devices can be added easily, and existing devices can be reconfigured or replaced, even while an application is running.
To allow optimal performance, applications are given direct access to graphics APIs, currently including OpenGL and Iris Performer [88, 89]. VR Juggler includes built-in support for performance monitoring of applications and graphics subsystems. It supports multiple-processor machines and will support distributing applications across multiple machines. Small base classes provide a skeleton for application development, while the abstractions of I/O devices simplify programming. During run-time, any VR Juggler application can be controlled or reconfigured by a Java-based graphical interface. VR Juggler is designed to support a wide array of VE hardware on a variety of architectures. Several tracking systems, gloves, and input devices are already supported. VR Juggler supports projection-based displays, and includes support for head-mounted devices.

1.3.3 AVANGO

Avango is a software framework which has been developed at the Department of Virtual Environments of the *German National Research Center for Information Technology (GMD)* [96]. This section is introducing the main concepts of Avango. A more specific description of the Avango C++ and scripting API is provided in chapter 5. Additionally the way distribution is carried out is discussed there.
Avango was designed for building interactive Virtual Environment applications for all kinds of display systems. Avango's design addresses the following requirements for a VE development system: performance, flexibility, rapid application development, extensibility, and portability. Applications in Avango are typically a collection of scheme scripts, which create and manipulate Avango objects and define relationships between them (Figure 1.9). Objects in Avango are field containers, representing object state information as a collection of fields, similar to Inventor [102]. These objects support a generic streaming interface for storing and retrieving an object and its state to and from a stream. This is the basic building block for object distribution. While objects in Avango are implemented using the C++ programming language, Avango also features a scripting environment used for customizing the

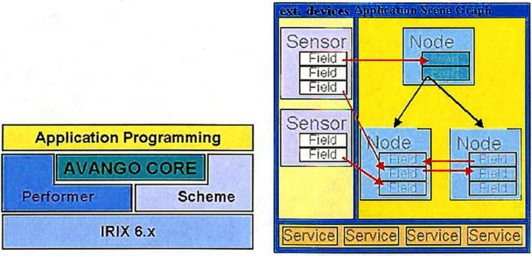


Figure 1.9: The left image shows the different modules on which the Avango core is positioned. The right images shows the Avango scene graph which consists of services, sensors and nodes.

runtime environment (e.g. setup of input and output devices) and developing applications. Avango includes the following concepts:

Viewer

All kinds of configurations of input and output devices can be assembled to viewers. The viewer is the interface between the user and the virtual world. Typical elements of a viewer are the visual, auditory and tactile displays as output devices and spatial trackers, audio and video sources as input devices.

Scripting

In addition to the C++ Application Programming Interface (API), Avango features a complete language binding to the interpreted language scheme. Scheme is an interpreted programming language descended from Algol and Lisp. It is a high-level language, supporting operations on structured data such as strings, lists, and vectors. This makes it possible to specify and change scene content, viewer features and object functionality and behaviour at runtime because all high-level Avango objects can be created and manipulated from Scheme. Script examples provided in chapter 5 given an idea of how applications are implemented with Avango.

Streaming

All objects are able to read and write their state from and to a stream. This is the basic facility needed to implement object persistence and network distribution. Persistence together with streaming support for objects make it possible to write the complete state of the system to a disk file at any time.

Distribution

All Avango objects are distributable, and their state can be shared by any number of participating viewers. Object creation, deletion and all changes at one site are immediately distributed to others. Details of distributed Avango are discussed together with the scripting interface in chapter 5.

Extensions

The system is extendible by subclassing of existing C++ system classes. This concerns objects as well as classes which encapsulate viewer features. Compiled extensions can be loaded into the system at runtime via Dynamically Shared Objects (DSO).

Interaction

Viewers provide input/output services which can be mapped to objects in the scene. Objects respond to events generated from input devices or other objects and deliver events to output devices.

Visual Rendering

Different displays have their appropriate rendering mechanism applied to the modelling hierarchy. Only the visual rendering has a direct access through the SGI Performer pipeline. The auditory and tactile rendering can be calculated on another computer connected to the master by a suitable network. The visual data processing is organized in a pipeline and computed in parallel by Performer. This rendering pipeline consists of a set of units, for:

- a database connection
- a user application
- the visual culling of the scene
- the intersection of objects
- the drawing of the scene

After the modelling hierarchy is updated to its actual state in the application process it is passed on to the culling process which strips all invisible objects. It is important to support this technique by dividing large geometry into smaller, cullable parts. The remaining part of the scene is passed on to the drawing process where it gets rendered on the screen using OpenGL. For configurations with more than one screen, the appropriate number of pipelines is used.

This chapter presented the basic technological components of Virtual Environments. The next chapter is focussing on interaction techniques and metaphors in VEs. It presents in detail related work to this thesis.

Chapter 2

Related Work

This chapter provides a survey on related work. In the first section many different interaction techniques are introduced, discussed, and compared with each other. It is shown which techniques that are used in two-dimensional graphical user interfaces (GUIs) are usable or portable to the three-dimensional Virtual Environments. Additionally new three-dimensional techniques are introduced. The last section deals with Collaborative Virtual Environments (CVE). It provides a survey on selected CVEs and tele-immersion applications. In addition interaction in these environments is discussed.

2.1 Interaction Techniques

Interaction techniques denote the possibility how different input devices are used in order to fulfill a special interaction task. *Interaction tasks* denote the entries of a unit of information by the user [39]. From the programmer's point of view interaction techniques can be interpreted as mechanisms with which the user inputs (events) are recognized and processed by the system. If user inputs consist of a sequence of different events, the interaction techniques are called complex according to [25, 26].
Basic interaction tasks are: selection, positioning, text input, numeric input, and confirmation [38].

When *selecting*, the user chooses an element out of a list of elements. The list is either of a fixed or variable size. In 2D GUIs selection can be implemented by a menu in combination with a mouse or a stylus. An alternative is to enter a name with a keyboard into a character input field.

For the *positioning* the user selects a point (x,y) in two dimensions, (x,y,z) in three dimensions and moves the selected object to this point. Positioning is usually performed by a locator device like the mouse or using a touch screen with a pen-like device as a stylus. Another possibility is to enter numeric coordinates using a keyboard.

Text input denotes the input and the manipulation of character strings. The keyboard is the dominant input device for text input. It is, however, also realizable using microphones and speech recognition software.

Numeric input (quantification) denotes the declaration of a value. It is usually performed using a keyboard. Sliders are an alternative if the numeric interval is static and not too large.

Confirmation is type independent and can be realized by any user input. Typically confirmation is performed by voice, by a pressed key of the keyboard or button of a special tool (mouse, joystick etc.). Confirmation mostly makes use of so-called command buttons. In non-graphical systems pressing the white space or the enter key is used in order to confirm actions.

2.1.1 Basic Interaction Techniques

Input and output devices are a very important factor influencing the choice of an interaction technique. For output devices as the monitor and the Head Mounted Display in combination with space mouse, pinch and data gloves many different interaction techniques are under current investigation [10, 13, 33, 46, 62]. Interaction techniques especially developed for rear-projection based displays like CAVEs and Workbenches in combination with, for example, pen-like stylus input devices, force feedback or even new input devices are seldom [33, 42, 56].
In the following sections different interaction techniques are introduced and discussed with respect to the completion of the five basic interaction tasks mentioned above. It is discussed which of these techniques are usable in Virtual Environments according to their advantages and disadvantages.

Command Languages

Command languages are the oldest interaction techniques. They enable a user to perform the interaction tasks: text and numeric input. To apply this interaction technique a keyboard is needed as an input device.
Command languages have a universal expressiveness. In common 2D applications and desktop Virtual Environments command languages cannot be used very efficiently. To use them efficiently is only possible for expert users as com-

mand languages are not easily learnable and they do not offer control outside their sphere of action. Another problem is that the user is forced to switch the communication modi when typing in commands for interaction.
The use of keyboards as input devices for command languages in most rear-projection based Virtual Environments is not very comfortable and satisfying. One reason is that the user is forced to turn the head towards the keyboard and away from the data set when entering commands. Another reason is the difficulty to position the keyboard outside the disposal when it is not needed. Especially CAVEs, Powerwalls and Cylindrical display systems would need to have extras racks in order to position a keyboard which then destroy the feeling of immersion.

Virtual Pick Rays

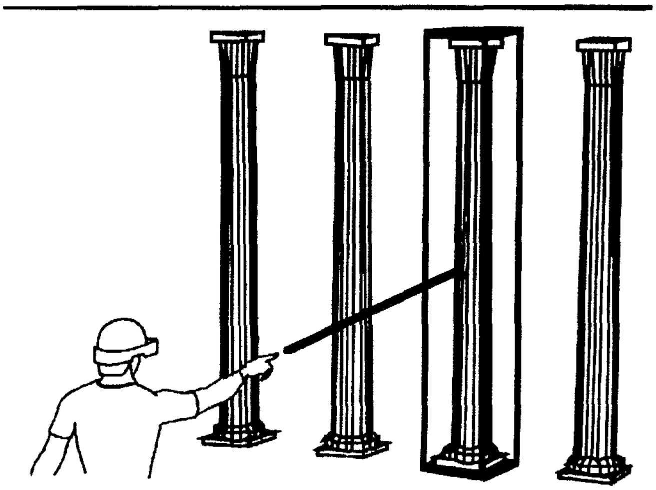


Figure 2.1: Interaction with a virtual pick ray. [Mine et al., [71]]

Virtual pick rays enable a user to select data from a distance. It can be used for selection, confirmation, and positioning.
The metaphor of using a ray is the three-dimensional analogon to the usage of the two-dimensional mouse pointer.
The pick ray interaction technique is difficult to use with 2D graphical user interface and desktop Virtual Environments since the user is sitting too close to the display system.
In rear projection based Virtual Environments as well as with Head Mounted Displays the virtual pick ray is an interaction technique that is easy to use in order to fulfill interaction tasks. For applying this interaction technique when working with these display systems input devices such as the stylus or gloves

can be used (see Figure 2.1).
Usually the interaction with the virtual pick ray such as selecting and picking is easy to learn and apply. A general disadvantage when working with a pick ray is the "lever arm" [71]. The longer the distance between the picked object and the picking device (normally a stylus in the user's hand) the bigger the movements of the object according to the movements of the hand. Interaction tasks which require high interaction precision might be insufficiently fulfilled using this technique. This is especially valid as long as the picked object is not directly attached to the manipulating hand or tool.

Menus

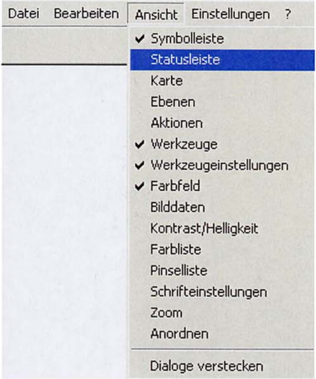


Figure 2.2: Example of a pull-down menu.

Menus enable a user to select from a list of choices. A menu consists of a list of items (icons, commands etc. according to [34]) where the size of the list is static (see Figure 2.2). They support the user's task through offering alternative actions. If the user selects the wrong element out of the list it is always possible to return and to begin the selection again. Menus can be permanently visible or they can be generated dynamically (pop-up and pull-down menus). If the list of choices is too big to show all alternatives at ones, the programmer

needs to think about other visual representations. This can either be done by partitioning the menu into another logically structured hierarchy or by representing the menu as a linear sequence of choices.
In the case of an hierarchical menu the user selects the first element from the beginning of the hierarchy. After that appears either the next list of choices or the choice itself (node of the hierarchy). An example is an hierarchical pop-up menu.
Typical WIMP (windows, icons, menus, pointing) applications usually provide a window and interface elements like menus, icons and toolbars to work on documents.
In two-dimensional graphical user interfaces pull-down menus are often aligned together with menu bars to the upper or lower edge of the screen.
General advantages in comparison to command languages are that the user only needs to visually recognize the information in a menu, whereas the user needs to recall information from its memory when using command languages. However, due to their complexity menu systems are to a certain degree contradictory to the demand for minimizing the search time while looking for a specific function or tool.
In Virtual Environments static menus have the disadvantage of limiting the interaction space and occluding parts of the data set. One of the first approaches which used pull-down menus in Virtual Environments is described by Jacoby et al. in [59]. More issues related the use of menus in Virtual Environments can be found in Darken's work [29]. In his thesis the focus is on visibility of menus and readability of fonts in VEs. In addition, guidelines and principles on menu placement in VEs are suggested by the author.
In Figure 3.6 of section 3.5 a so-called ring-menu is introduced which is developed within this thesis. It is similar to Liang's ring menu [68]. One ring menu is assigned to one data set. When a user asks the data set for its functionality the menu appears and shows a list of choices representing operations that are applicable to the data set. The menu is attached to the user's hand position. It only follows the translation of the user's hand whereas the rotation of the user's wrist is used to intersect the "cake pieces" with the pick ray. Thus selection of operations is possible. The advantages are that the ring menu is generated dynamically and only exists visually as long as the user needs it. As the menu is attached to the hand the user does not need the change the viewpoint towards the menu and away from the data. Unfortunately this involves that the ring menu is aligned between the user and the data set as it is used in a three dimensional space. For not occluding parts of the data the menu is designed to be almost transparent.

Buttons

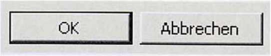


Figure 2.3: Example of a button pair.

Buttons are usually represented by oval graphical forms including text or icons. Pressing a special mouse button above these buttons enables a user to execute certain actions.
There exist different types of buttons: command buttons and check buttons. Command buttons are mostly used for confirmation purposes as the interaction task. Also high order commands and other procedures can be executed using command buttons.
Check buttons represent the states TRUE or FALSE. They are switchable using the mouse click. These check buttons represent a possibility for the selection between two choices.
Buttons can grouped together to button blocks (see Figure 2.4). An exclusive

Figure 2.4: Button block with check buttons.

button block for example consists of some check buttons from which only one represents the state TRUE at a time. Invisible buttons which are positioned above other graphics are called hot-spots. With the help of those hot spots interactive graphical objects are simulated.
Buttons are usable in 2D as well as in 3D applications. In Virtual Environments for example buttons can be pressed using virtual pick rays.

However, the interaction designer has to carefully think about where to position the button in a way that they do not occlude parts of the data.
Especially in Virtual Environments the problem is the graphical representation of the buttons. Buttons represented by two-dimensional icons are smaller and do not reduce the limited interaction space as much as three-dimensional icons would. However, for the user who tries to intersect the button with a pick ray from arbitrary positions in the view frustum three-dimensional buttons are much easier to hit. As a consequence the disadvantage is the reduction of the view frustum.

Toolbars

Toolbars enable a user to customize and manipulate applications or data sets while selecting tools from a list of choices.
Toolbars denote a very common interaction technique and are used in two-dimensional GUIs as well as in three-dimensional Virtual Environments.
Even two-dimensional toolbars are possible in a Virtual Environment. However, here occurs the same problem as already discussed for the buttons and menus. Is the toolbar positioned fixed in front of the user at the RWB for example a two-dimensional representation is conceivable.
Within this thesis a toolbar has been designed and implemented which allows the user to select generic operations and apply them to the data set (see Figure 3.7 in section 3.5). This toolbar is positioned in front of the user and thus does not occlude any part of the data. A button block with five buttons on top is chosen as geometrical form for the toolbar. Each button represents another operation, such as drag, zoom, rotate etc..
The basic operations are positioned so that they are always visible and accessible. This supports the work flow since they are frequently used during the interaction with the Virtual Environment.

Sliders

Figure 2.5: Slider.

Sliders enable a user to determine a value within a certain interval. They are used for quantification. Scroll bars are a special implementation of sliders. With their help it is possible to change the visible part of an in principle infinite

screen.
They are used in window-based systems and graphical editors. Sliders can also be used for interaction tasks in Virtual Environments. But the representation of the sliders presents the same problem as it did for the buttons.
The interaction designer needs to decide whether a two-dimensional or a three-dimensional representation is preferred.

Scrollable Selection Lists

Figure 2.6: Scrollable selection list.

Scrollable selection lists enable a user to select one or more items from a list by clicking them with a mouse. They support the selection interaction task.
The size of scrollable selection lists is independent from the number of items in the list in contrast to button block and menus. With the help of the slider the user is able to access even invisible entries.
In Virtual Environments the usage of scrollable selection lists is possible but very uncomfortable and offers poor usability.
Alignment and graphical representation similar to the ones of the character input fields and buttons present the major problems when trying to use the selection lists directly in the view frustum. Scrollable selection lists on wireless touch screens or PDAs (Personal Digital Assistant like the Palm) are an alternative.

Character Input Fields

Character input fields enable a user for text input. Mostly only single characters are entered. More complex character input fields are multi-line fields with scroll bars for scrolling through the text.
Character input fields are usable in desktop based Virtual Environments with a monitor as output device and a keyboard as input device.
The keyboard is a very uncomfortable input device when combining it with rear-projection based Virtual Environments. In VEs it is better not to use a keyboard for character input. In these VEs it is possible to use wireless touch screens and PDAs with a pen for text input. These input devices do not deliver the discussed keyboard problems but they need a place to store them when they are not used.
At a Responsive Workbench the storage of these input devices does not present a problem and their usage is possible.
In a CAVE only the usage of small PDAs is possible which are storable at the user's body.

Masks and Forms

Masks and forms provide a user with static possibilities to enter information two-dimensionally.
Masks are two-dimensional indicators of the systems state which occlude the whole display screen and consist primarily of alpha numeric characters [31]. In static areas information is represented about different system states, inputs and system messages.
The work area in which the user can enter information usually is designed as a form. Forms are two-dimensional alignments of fields in which information is perceived and entered [31]. They are very suitable for grammatical formulation of orders through the combination of specifications as well as for meta communication since they are able to represent information about the system state.
In two-dimensional graphical user interfaces masks and forms are usable as long as the system state information has to appear and stay in the foreground of the disposal.
In Virtual Environments masks and forms create the same problems as menus and scrollable selection lists. Their size even aggravates the poor usability.

Dialog Boxes

Dialog boxes enable a user to execute complex operations which need a lot of parameters. They support the interaction tasks: text and numeric input,

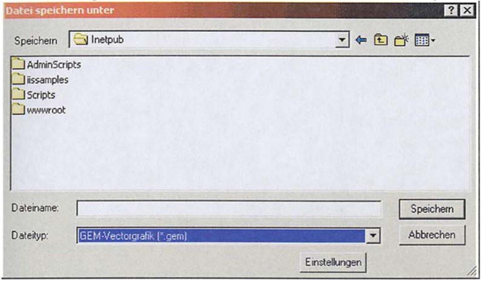


Figure 2.7: Example of a dialog box.

selection and confirmation.
Generally a dialog box consists of other elements like, for example, check buttons, numeric input fields, an OK-button for confirmation and thus approving the parameter set, or a Cancel-button for cancellation and thus ignoring the parameter set.
In either case the dialog box is closed and the dialog ends. Dialog boxes are used in Virtual Environments too.
Figure 2.8 shows a VE designer interacting over a dialog box in a virtual museum application. This VE dialog box consists of check buttons and numeric input fields. The interaction is performed using a pen-like stylus in the user's hand. For not occluding data behind the box is designed to be almost transparent inside the frame.

2.1.2 Advanced Interaction Techniques

Speech Recognition

Speech recognition is a non visible interaction technique. It enables a user to perform actions by the voice which would be far more complex to perform with other techniques.
Speech recognition contributes to user comfort and is dimension independent. This means that it can be used as interaction technique in two-dimensional applications as well as in three-dimensional Virtual Environments as a supplement to hand based interaction techniques.

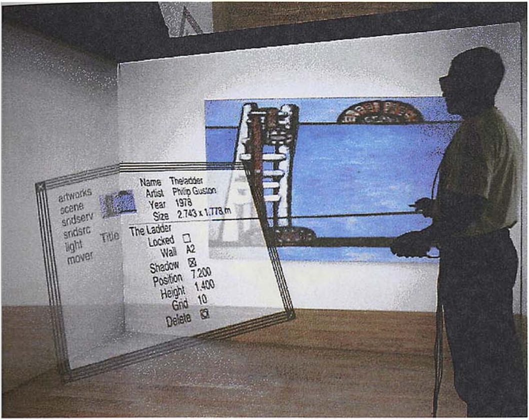


Figure 2.8: This is an example of a dialog box in a CAVE Virtual Environment. Check buttons and numeric input fields with sliders are used to enter information. In order to not occlude data the dialog box is designed to be almost transparent inside the frame.

A general problem with speech recognition software is the recognition rate. Speaker dependent speech recognition software toolkits have a much higher recognition rate. Speaker dependency, however, implies that the speaker has to create its own database with spoken phonemes.

In Collaborative Virtual Environments the interference of speech recognition with the oral communication between the remote users causes problems.

One approach to overcome this problem would be to implement a code word like "computer", which switches the system into the "Listen" mode as soon as it is recognized. The communication with the remote partner is muted and the computer waits for aural commands of the user.

Combinations of voice input with three-dimensional localized sound and two-handed manipulation in Virtual Environments are described by NASA Telep-

resence research in [36].
Systems which offer a method to embed voice communication in Virtual Environments by means of voice annotations are proposed by Harman et al. and Verlinden et al. in [52, 99]. These systems allow a set of capabilities for inserting, iconizing, playing back and organizing annotations in VEs.

Tactile/Force Feedback

Tactile and force feedback interaction techniques enable a user to touch and feel computer generated data. Tactile techniques provide information about surface attributes as a feedback to the user [82]. Force feedback techniques provide information to the user that are related to the object's mass. These techniques can be used as a supplement to visual perception [104].
Haptic interaction techniques are mostly used in desktop based Virtual Environments or together with ReachIn display systems (see section 1.1.2).
In rear-projection based Virtual Environments it is problematic to mount the feedback device when using force and tactile feedback interaction techniques [19].
When working at the Responsive Workbench with a Phantom the force feedback device is mountable as shown in Figure 1.3. If force feedback is not further needed the Phantom arm can be positioned outside the view frustum and thus does not disturb the user during the task.
Different types of haptic and tactile feedback are especially interesting for applications which have an high demand for precision. Examples for force feedback applications are flight simulators with motion platforms or force feedback driven surgical simulators.
A comprehensive introduction and reference on force-feedback can be found in Burdea's work [22].

Direct Manipulation

Direct manipulation enables a user to manipulate data sets. It groups together all interaction techniques which deal with the "handling of objects".
Metaphors dealing with the perception of two-dimensional faces and three-dimensional spaces build the basis for direct manipulation.
The main principles of direct manipulation are:

- permanent visibility of the objects to be manipulated
- permanent visibility of the actions applied to the objects
- substitution of complex commands by physical actions such as movements with an input device

- fast and reversible user interaction with direct and mostly visual feedback.

Advantages of the direct manipulation are:

- beginners can quickly learn to perform direct manipulation
- occasional users can easily recall interactions
- experts can work efficiently
- support of systematic work due to direct feedback of interactions
- low user fatigue due to easy comprehensibility, predictable system reaction and possibilities of undo operations.
- error messages are far less important for direct manipulation than for other interaction techniques.

Nowadays, direct manipulation is the most used interaction technique due to its naturalness. The computer vanished to be only a device for experts mainly because of the development of direct manipulative graphical user interfaces.
In Virtual Environments direct manipulation is the most used interaction technique and many other techniques are making use of the direct manipulative metaphor [9, 37, 54, 70, 77]. The metaphor is the basis for the development of body-relative interaction techniques which are presented in the following.

2.1.3 Body-relative Interaction Techniques

Body-relative interaction enables a user to perform actions relative to the own body. It is not a real interaction technique but rather an additional criterion for other interaction techniques.
Body-relative interaction techniques are more effective than interaction techniques relying solely on visual information as they provide a physical real world frame in which to work and they provide a more direct and precise sense of control.
During body-relative interaction a user can take advantage of *proprioception*. Here *proprioception* denotes the user's sense of position and orientation of his body and its several parts [7]. Proprioception is used again in Awareness-Action-Feedback loops of the CVE interaction taxonomy developed in this thesis (see section 3.6).
Body-relative interaction is proposed by M.Mine and used by other VE interaction designers too [54, 71, 72].
Body-relative interaction is especially important for the direct manipulation.

An example for the combination of different interaction techniques supported by body-relative interaction is constructible for a surgeon simulating an incision. The surgeon who is concentrating visually and tactilly/haptically on the incision turns the hand upside down, opens it, and calls the name of the surgical tool that is needed next.
This complex interaction task includes gesture recognition (turning and opening the hand using proprioception as he does not need to raise the head and look), speech recognition (calling the tools name) and body-relative direct manipulation (closing the hand with the tool and applying it to the data set in front). Although this interaction is very complex it is easy to learn as it exploits a real-world knowledge metaphor.

Gesture Recognition

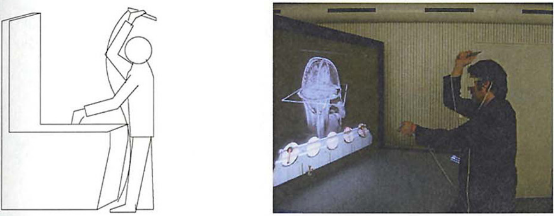


Figure 2.9: The user deselects the current tool by throwing it over the shoulder.

Gesture recognition enables a user to perform actions without any auxiliary tools. The user's gesture recognized by the system can be interpreted as selection, confirmation and positioning, including scaling, rotation and translation. Comparable with speech recognition the gesture recognition interaction technique can easily combined with other techniques that make use of input devices. Gesture recognition interaction techniques are easy to apply in desktop and rear projections based VEs. Usually a video camera is recording the user during the interaction and analysing hand and head movements in real-time [66]. Other approaches make use of the user's input devices that are usually tracked already, such as through the location sensor attached to the shutter glasses and stylus or pinch glove.
Examples for the gesture recognition based interaction techniques are the so-called "over-the-shoulder deletion" and the two-handed flying proposed by

M.Mine [70, 72] (see Figure 2.9).
The system recognizes the movement of hand over the shoulder as the command to deselect the current tool from the input device in the user's hand. In the other case the system interprets the position of the user's two hands relative to each other as the vector which determines the navigation direction. The relative distance of the hands corresponds to the absolute value of the vector and is recognized as the velocity of the movement.
In another application gesture recognition is used to facilitate the interaction

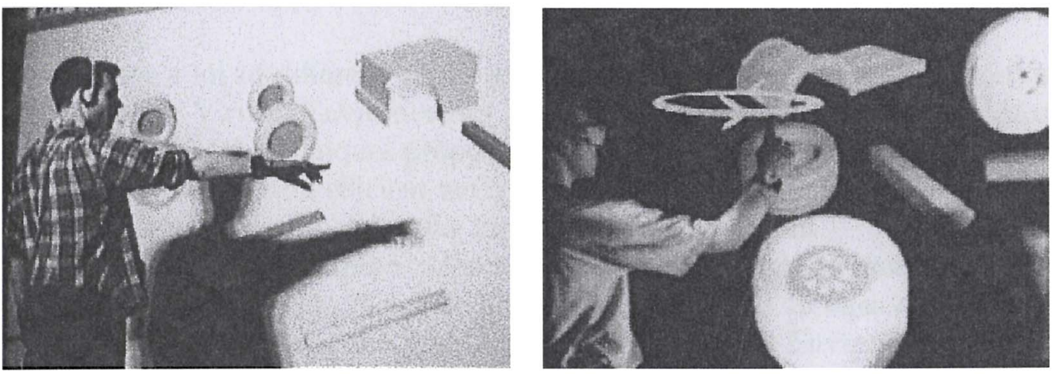


Figure 2.10: Natural speech and gesture based input used for virtual assembly simulations. In the figure on the right hand side a gesture is used for describing an object rotation. [Wachsmuth et al., [61]]

during a virtual assembly simulation as for example in [61, 100, 101]. Here the gestures are recognized as commands for "rotate", "drag", "translate" etc. (Figure 2.10).

Arm Extension

The extended arm interaction technique enables a user to select, grasp and position distant objects without navigating there. The arm extension is a pure three-dimensional interaction technique.
An example is the so-called "Go-Go" technique proposed by Poupyrev et al. and used by many others [9, 11, 77].
The "Go-Go" technique enables a user to interact within arm distance but also with distant objects deeper in the virtual scene. Therefore the position of the arms is under permanent control. A non-linear function allows the virtual arms to "grow" if needed. For this the user performs a special hand movement. If the user wants to return to the normal arm length a corresponding hand movement shortens the arms again.
Although this technique is especially developed for Head Mounted Display

based Virtual Environments it can easily be applied to rear-projection based Virtual Environments too. Applications can be virtual landscape architecture, urban planning or interior design.

2.1.4 Other Techniques and Interaction Frameworks

In combination with other techniques such as speech recognition new and intuitive three-dimensional interaction techniques are designable. The more promising ones of these techniques are presented in [9, 11, 16, 17, 18, 36, 52, 58, 77, 99].

There exist a lot of different works mainly dealing with travel and navigation techniques in Virtual Environments [33, 35]. Among them are the *Worlds in Miniature (WIM)* techniques proposed by Stoakley et al. in [93]. The focus of that work rather lies on efficient navigation and orientation, being independent of the specific activities and tasks in a Virtual Environment.

An approach that attempts to develop a taxonomy of interaction techniques can be found in [10, 11, 15]. The authors Bowman and Hodges developed a methodology to increase the usability aspects of Virtual Environments using Head Mounted Displays.

They tried to facilitate the design and evaluation of interaction techniques in VEs with this taxonomy. They distinguish between different classes of interaction techniques. These are considered to be viewpoint changes (navigation, travel), selection and manipulation.

However, their approaches are restricted to fully-immersive environments using Head-Mounted Displays. The influences of other display systems on the interaction in a Virtual Environment are not considered. In addition, their taxonomy is not able to explain the impact of interaction feedback, Virtual Environment representations, and awareness factors on usability aspects.

Conclusions

Anyway, it is possible to conclude that there are interaction techniques which are applicable in two-dimensional Graphical User Interfaces (GUIs) as well as in three-dimensional Virtual Environments (VEs). However, a direct usage of two-dimensional interaction techniques in three-dimensional VEs is not always possible. Most of the techniques have to be modified to be able to fulfill three-dimensional requirements. These requirements are for example accessibility from arbitrary points in the view frustum without occlusion, direct and precise sense of control, as well as intuitive and effective use. There are for sure many techniques available which tend to address all these requirements but new techniques especially for Virtual and Collaborative Virtual Environments have to be developed. A good reference frame for doing this represents the body-relative interaction even for rear-projection based Virtual Environments.

The introduced interaction techniques are presented according to their usability in Virtual Environments but not especially according to their usability in Collaborative Virtual Environments (CVE). The next section is presenting an overview of CVEs and explaining how the collaborative aspect in these environments is carried out.

2.2 Collaborative Virtual Environments

2.2.1 CVE applications

CVEs are multi-party Virtual Environments which allow a number of users to share a common virtual space, where they may interact with each other and the environment itself.
The problems of multiple users sharing the same workspace are already known from the field of Computer Supported Collaborative Work (CSCW) and groupware [4, 32, 57]. Some of the major problems are: the distribution of objects [98] and information as well as the delegation of rights and the representation of group structures.
Interest in *spatial approaches* to CSCW has grown over recent years. Specific examples of the spatial approach include media spaces [6], spatially oriented video conferencing [55, 57, 83], Collaborative Virtual Environments [3, 95] and tele-presence systems [66].
In contrast to CSCW systems, direct collaborative real-time interaction leads to completely new interaction possibilities, especially concurrent interaction of at least two users with one or more objects [73]. Unfortunately there is a lack of application and design support for CVEs.
In addition, most of the CVEs under investigation are web based collaborative Virtual Environments. Good overviews about examples of these desktop CVEs can be found in the following literature [3, 4, 21, 40, 69, 92, 95].
There exist only a few approaches of back projection based VEs and CVEs. Overviews about these approaches can be found in [20, 28, 43, 81].

2.2.2 CVE evaluations

When designing and implementing collaborative applications it is not sufficient to consider only technological aspects of the application. Usability as well as interactive qualities of the implementation are at least as important.

For measuring usability aspects of applications there exist four different approaches [79]. These are the:

- Interaction oriented approach
- User oriented approach
- Product oriented approach
- Formal approach

The *interaction oriented approach* is the most common one and is concerned with all kinds of usability testing with users. The *user oriented approach* tries to measure usability quality in terms of mental effort and attitude of the users. This is done by using questionnaires and interviews and is mainly used in this thesis (see chapter 6). The *product oriented approach* is concerned with measuring ergonomic attributes and thus quantitative measurements are necessary. Lastly the *formal approach* tries to simulate usability in terms of formal models [75]. This approach can be seen in the context of theory based evaluation.

Although usability testing is very important only a few authors presented evaluation studies for designed Collaborative Virtual Environments. On the other hand, existing evaluations try to draw results from assessments of 4-10 experimental subjects. The reasons are that user evaluation is a really hard task and that it is difficult to find evaluators.
In addition, many existing evaluations are focussing on very different aspects of VEs or CVEs. So for example, were Witmer and Singer one of the first who tried to measure the degree of immersion [103]. Therefore they developed a very detailed questionnaire which, in the research community, is discussed controversially [90].
Slater et al. are also trying to quantify the degree of immersion and presence [91, 92, 97]. Their approaches and experiences are restricted to desktop Virtual Environments using a normal monitor and DIVE as the software framework but no fully- or semi-immersive rear projection-based Virtual Environments.
Other usability evaluations are focussing on interaction devices and techniques as presented by Beaten and Dehoff [2] and Bowman et al. in [14]. Evaluations assessing the usability of Virtual Environment's user interfaces are presented by Hix et al. in [54]. These evaluations are restricted to stand-alone Virtual Environments using a one-sided Responsive Workbench as display system. Unfortunately the usability findings are only focussing on battle-field simulations.
The goal of this thesis is to provide design, application development and interaction support for Collaborative Virtual Environments. This includes that

the approach developed in this thesis provides means to extensively evaluate and assess the usability of CVE user interfaces and collaborative awareness factors [50].

In the following a survey on selected important CVE systems and applications is presented. The particular reason for choosing these CVE systems is the fact that they specifically address the needs of users working cooperatively together in a Virtual Environment.
Teleport and the National Tele-Immersion Initiative (NTII) uses effectively tele-presence techniques to enable face-to-face communication in an augmented office environment.
NICE for example addresses the networking aspect as well as the problems of shared manipulation. Our CVE goes one step beyond this while bringing together tele-presence and collaborative rather than cooperative interaction allowing for shared manipulation of data while preserving face-to-face communication. The following sections are presenting the mentioned CVE systems in more detail.

2.2.3 Teleport

Nowadays personal computers equipped with microphone, speakers, camera, and perhaps additional video monitors, are widely used for desktop video conferencing. Conference participants appear in windows, or on adjoining monitors, and may access shared applications shown simultaneously on each participants' screen. Several desktop video conferencing systems have been described in literature and commercial products are available. But while desktop video conferencing has certainly been shown to be useful for a variety of tasks and has many advantages when compared to earlier forms of video conferencing involving special meeting rooms, it is still recognized that there are many situations where desktop video conferencing is not appropriate.
The Teleport environment is designed to overcome the disadvantages of desktop video conferencing and to establish life-like conference sessions that bring people together as if face-to-face (see Figure 2.11).
Teleport has been developed at the Department of Virtual Environments of the *German National Research Center for Information Technology (GMD)* [20, 45]. The system is based around special rooms, called display rooms, where one wall is a view port into a virtual extension as shown in Figure 2.11. The geometry, surface characteristics, and lighting of the virtual match the real room to which it is attached. When a teleconferencing connection is established, video imagery of the remote participant is composited with the rendered view of the virtual extension. The viewing position of the local participant is tracked,

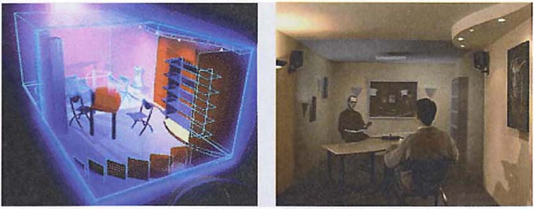


Figure 2.11: Teleport Display Room.

allowing imagery appearing on the wall display to be rendered from the participant's perspective. The combination of viewer tracking, a wall sized display, and real-time rendering and compositing, give the illusion of the virtual extension being attached to the real room. The result is a natural and immersive teleconferencing environment where real and Virtual Environments are merged without the need for head-mounted displays or other encumbering devices.
The system uses a 3x2.25m rear-projected video wall attached to a $3m^2$ room. The video wall is driven by a pair of high luminosity video projectors. Both projectors can display mid-resolution video signals and high-resolution RGB signals. A camera is placed on a stand or table and set approximately at eye height. The field of view is wide enough to take in a full upper body shot of the local participant. The viewer tracking system determines the position of the local participant within the display room, from which the viewpoint is derived.
Chroma keying is used for segmentation in order to determine the regions in the video images where the participant appears. The virtual extension is rendered from the viewpoint of the tracked participant located in the display room. Because this person is free to move within the display room, the virtual extension must be continuously re-rendered. For audio, each participant wears a small microphone. The audio streams from remote participants are mixed together and sent to speakers mounted on either side of the video wall.
The Teleport system provides video conferencing integrated into a room environment as its best. However, the participants are restricted to viewing and talking to each other. 3D interaction is not implemented at all and thus sharing of three-dimensional data is not possible.

2.2.4 Tele-Immersion Initiative

The National Tele-Immersion Initiative (NTII) is an American research community investigating into tele-presence applications together with internet2. Internet2 is a consortium led by over 180 universities working in partnership with industry and government to develop and deploy advanced network applications and technologies. Internet2 is recreating the partnership among academia, industry and government that fostered todays Internet in its infancy. The aim of NTII is to enable users at geographically distributed sites to collaborate in real time in a shared, simulated environment as if they were in the same physical room.
The participating research sites can be found under the following URL:
http://www.advanced.org/teleimmersion.html
There exist plenty of different tele-immersion approaches. The sketch from University of North Carolina at Chapel Hill in Figure 2.12 provides an idea about the research activities of NTII [78].
Currently the partners within the NTII consortium are able to do stereo-video

Figure 2.12: The figure sketches a vision of the office of the future. Current tele-immersion systems try to support distributed collaborative work. [Fuchs et al.,1998]

conferencing in real-time. In addition, very rudimentary collaborative interaction support is enabled. The interaction technique implemented so far is the direct manipulation with pick rays. Therefore, until now, the implementation of the office of the future can be seen as a spatial video conferencing tool rather than a collaborative office with remote access to all kinds of information.

2.2.5 NICE

NICE is the acronym for *Narrative-based Immersive Constructionist / Collaborative Environments*. It is a prototype of an educational, distributed Virtual Environment for young users using CAVEs as displays. NICE has been developed at the University of Illinois at Chicago by M. Roussos et al. [60, 80, 81]. The main constructive activity in NICE is to build and develop small local ecosystems on the bare parts of an island. The terrain serves as an open land which the child explores and decides where to plant and populate.
Various seeds for planting garden vegetables and trees are stored in crates and serve as starting points for building micro-ecosystems on the island (see Figure 2.13). When the user drops a seed on the ground, the corresponding plant, flower or tree will start to grow. The pace in which this happens can be predetermined; the user may choose to see the system grow very quickly, or, in the case of a school project, extend it over the period of a semester. The tomatoes, carrots, pumpkins and other plant objects contain a set of characteristics that contribute to their growth. They all have values for their age, the amount of water they hold, the amount of light they need, their proximity to other plants of their kind. These values determine the health of the plant and its size. Visual cues aid the child in determining the state of a plant or flower. When the cloud has been pouring rain over it for too long, the plant opens an umbrella, when the sunlight is too bright, it wears sunglasses.
Sound in the environment enriches the surroundings in a variety of ways. Dif-

Figure 2.13: Children, and remote children represented by avatars, plant a garden in the *NICE* distributed Collaborative Virtual Environment. [Roussos et al., [80]]

ferent environmental sounds are experienced depending on where each participant is standing, e.g. the children by the shoreline will hear the water, the children in the rainforest will hear the birds, etc.. The user interacts with the Virtual Environment using 'the wand', a simple tracked input device containing a joystick and three buttons. The CAVE's roomsized structure allows for multiple users to move around freely, both physically and virtually surrounded by vivid displays. However, only one person is being tracked in each CAVE, thus reducing the participation of all the children in the same CAVE at one time. One child can set the viewpoint, another child may handle the wand to pick and place objects, while the other children can observe and give their verbal input. It was observed that this has not decreased the feeling of presence and immersion of the participants that are not tracked and the children may exchange roles at any time.

The system is developed in a prototype system named GULLIVR Graphical User Learning Landscapes In VR. Specifically GULLIVR allows multiple participants to share in the exploration of a virtual space, interact with each other, and perform simple tasks. A simple agent architecture based on a frame system, is being developed to support the actions and personalities of the characters. The core of GULLIVR is the CAVE library. On top of this GULLIVR uses SGI Performer and OpenGL to render the Virtual Environment. Although GULLIVR was originally conceived for the CAVE, it is capable of supporting a number of different VR platforms including the Responsive Workbench, and simple graphics workstations. GULLIVR's network component allows multiple networked participants to explore the same virtual space. Multiple distributed GULLIVRs running on separate VR systems are connected via a centralized database server that guarantees consistency across all the separate environments. The communications library supporting GULLIVR is based on a client/server model where the number of remote clients is limited only by bandwidth and latency.

The wand is the physical interface to the virtual world. It is used to navigate around the virtual world, and to manipulate virtual objects within that world. The user can move around within the boundaries of the CAVE, walking around or through virtual objects, and can press the joystick on the wand to move the CAVE through the Virtual Environment. GULLIVR provides the option of flying over the world, or adjusting the floor of the CAVE to coincide with the height of the landscape, thus allowing the user to climb over terrain or ascend and descend stairs, by physically walking in the CAVE. Every element of the scene in GULLIVR is treated as an object. Hence, every object serves as a building block for the construction of other objects. The child can pick up objects in the Virtual Environment by using the wand as an extension of her hand. She moves the wand over to an object and clicks a button on the wand to pick it up. She can then move the object to an appropriate place and let it go.

The NICE application belongs to the most advanced CVEs. However, the system design restricts NICE to be an edutainment application for kids or an artificial basis for game applications and story telling. Artificial avatars are used instead of real looking video representations of remote participants. In this context, view point control and consistence as well as the alignment with distributed data is not implemented as it is of minor importance for the users. In addition, the used techniques allow simple interaction especially designed for kids planting a garden co-operatively.

2.3 Conclusions

This chapter described related work to the thesis. It is shown that there are interaction techniques which are applicable in two-dimensional Graphical User Interfaces (GUIs) as well as in three-dimensional Virtual Environments (VEs). However, a direct usage of two-dimensional interaction techniques in three-dimensional VEs is never possible. Most of the techniques have to be modified to be able to fulfill three-dimensional requirements. Hence, new techniques especially for virtual and collaborative Virtual Environments have to be developed. A good reference frame for doing this represents the body-relative interaction even for rear-projection based Virtual Environments. In combination with other techniques such as speech recognition new and intuitive three-dimensional interaction techniques are designable.
The remaining sections are a survey on selected important CVE systems and applications. They provided an idea about the use of tele-immersion approaches and Collaborative Virtual Environments. From the basic chapter 1 and during the presentation of the different CVE approaches it becomes clear that the absence of a classifying framework for collaborative interaction is the most challenging problem when designing VEs and CVEs. Hence, in the next chapter 3 the goal is to develop an own approach with which designers and programmers are able to analyze user tasks and interaction cycles in VEs. Then, chapter 4 describes an application for a pre-described two user task scenario. Chapter 5 discusses implementation details and in chapter 6 the developed applications are evaluated assessing usability and collaborative awareness.

Chapter 3

Interaction Framework

The goal is to provide an Interaction Framework to facilitate and guide the design of Collaborative Distributed Virtual Environment applications. This thesis focuses on projection based Virtual Environments (VE), where the collaboration can reach immersive face-to-face communication with the computer being the transparent medium. In this chapter a Human-Computer-Human (H-C-H) model and an H-C-H interaction framework is presented. This model is first used to design a CVE in chapter 4 and is evaluated in chapter 6.

3.1 The Human-Computer-Human Model

Currently the various modalities of interaction between humans and computers are under investigation in the *HCI* and *CHI* research community. Unfortunately in most applications the Computer-Human interaction is discussed separately from the Human-Computer interaction. Applications where these topics are addressed together are mostly groupware applications [8, 32]. In order to discuss *HC* and *CH* interaction together a Human-Computer-Human Model is proposed. Within this model the following interactions are observed:

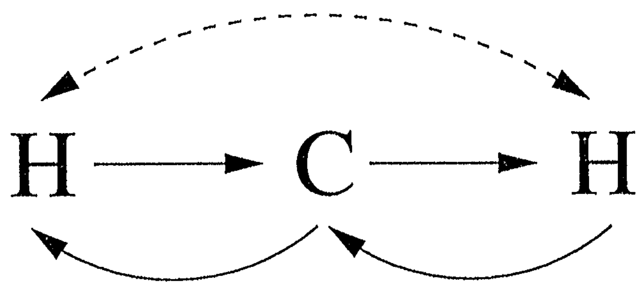


Figure 3.1: Interaction within the Human-Computer-Human Model. This model unifies the various modalities of interaction between humans and computers.

- $H \rightarrow C$ *Flow* : User input
- $C \rightarrow H$ *Flow* : Computer output
- $H \rightarrow H$ *via* C *Flow* : Inputs by one user are going to be displayed by the computer as outputs to the other user.

It is obvious that *direct* human to human interaction cannot happen in distributed environments. Especially when interacting with another user within a Virtual Environment the computer has to mediate and regulate this interaction. The computer as a mediator needs to set up a communication link like a video/audio connection and needs to manage the user's input and provide it as output to the other user. This follows from the fact that the interaction between human and computer is of a bidirectional nature. The computer waits and reacts on the users input queue, processes this input and then displays the reaction appropriately.
Now when looking at Figure 3.1 again it becomes clear that the Human-Human interaction can be described completely by a reduced bidirectional $H \leftrightarrow C$ chain. Although this chain is a subset of the *H-C-H* model it is possible to see that the computer processes similar inputs at both sites and presents the corresponding outputs to both sites appropriately. In other words the computer has to become omnipresent but transparent to the users.
The users of Virtual Environments do not need to know what computer hardware, what software or what communication mechanisms generate the VE. The interfaces at both sites need to be capable of hiding these from the users. As the computer becomes invisible the users have to have the feeling of communicating and interacting with another person as if face-to-face assuming an appropriate representation form for the remote user. Therefore the goal is to design distributed Collaborative Virtual Environments with interfaces in a way that the $H \leftrightarrow H$ interaction can happen as effective as possible.
As the computer is the mediator and regulator for this Human-to-Human interaction, C has to be optimized. The way to do this is optimizing the user interfaces as the interfaces for the $H \rightarrow C$ inputs and optimizing the viewer as the interface for the $C \rightarrow H$ outputs. In order to optimize these interaction flows within the *H-C-H* model, influence factors should be determined. These influence factors are responsible for making a user feel immersed within a Collaborative Virtual Environment and being able to communicate and interact with another person as if face-to-face. To consider these influence factors that come in play when creating distributed, Collaborative Virtual Environments, a taxonomy[1] of these influence factors is developed [47, 49, 50].

[1]The term "Taxonomy" is derived from the Greek word taxon. It is used in the science of biological classification. The taxonomist creates from a varied array of organisms a hierarchy

This developed taxonomy is not a hierarchy of classified VEs or CVEs but

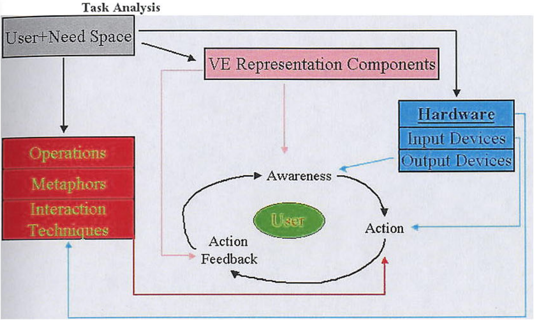


Figure 3.2: The VE and CVE design model.

a hierarchy of classified influence factors that show an impact on the design process of VEs and CVEs (see Figure 3.2). With the help of this taxonomy of influence factors it is then possible to develop a VE/CVE design model which supports the VE/CVE designer to consider the large amount of these influence factors. Thereby the design model also shows the dependency of the influence factors and enables to simulate the appearance of the VE/CVE early in the design process. The input for this simulation delivers the requirement engineering process that uses the task description and task analysis which determine the User+Need Space. An example for such a simulation is given in section 3.8.

The reason for the development of a new VE/CVE design model instead of taking an existing one is that the existing models developed in the CSCW and HCI community do not pay enough attention at Human-to-Human communication and collaboration in large scale projection based Virtual Environments. Bowman et al., for example, developed a taxonomy of different techniques

---

of groupings that have an orderly relationship to each other. The term is also used by Hix et al. [44] and Bowman et al. [11] for the categorization of usability characteristics and interaction techniques.

concerning the three tasks navigation/locomotion, selection and manipulation. This taxonomy enables the VE designer to find the interaction technique best-suited for a given task. The orderly relationship between the classified techniques is obtained according to usability issues taking into account user input devices, tasks and others.
The taxonomy has been developed for and evaluated in HMD based systems (Head Mounted Displays) but not in projection based displays. Whether this taxonomy is still valid for projection based display systems has to be proven in future.
As already mentioned, the existing taxonomies and VE design models do not consider collaboration and human-to-human communication aspects sufficiently which makes them useless for this work. Hence, it justifies the development of a new taxonomy of influence factors and with this the development of a new VE/CVE design model.
Therefore the utility of the taxonomy is considered rather than its absoluteness or completeness. The objective is to facilitate guided design of applications for supporting team work in CVEs [47].
One way to verify the generality of the approach is through the process of categorization, which allows to understand the low-level makeup of interaction techniques. This categorization may also lead to new design ideas. For designing the correct VE and choosing the most appropriate interaction technique using the H-C-H model, first the user tasks are specified and analysed as described in the following section. This task analysis determines the User+Need Space.
Also software engineering approaches typically use task analysis methods [30], which are concerned with the entities and the processes or tasks that need to be implemented in software. In a similar manner the User Task Analysis in this approach is concerned with the virtual data and representation components which can be considered analogous to the entities. Since this User Task Analysis is primarily focussing on collaboration and human-to-human communication, the specification of the VE/CVE is composed of the awareness-action-feedback loops and the metaphors, operations and interaction techniques which are implementing them.

3.2 User's Task Description and Analysis

Figure 3.3 shows that the approach starts with a *User's Task description (UTD)*. For example a task description could be:

Assume two users who want to connect two wooden laths. They use two ham-

mers and a box of nails. For pulling nails that are wrong pound into the wood they use a pair of pliers. They stand on top of a roof. Both laths have firstly to be connected so that they tower above the roof. Then they have to be attached to the roof. The laths are very heavy and can only be handled using both hands. The interaction space on the roof is reduced to two square meters. One user holds the wooden laths and the other user pounds the nails with the hammer or uses the pair of pliers. The users carry their tools (hammers and the box of nails) because there is not enough place to deposit them on the roof.

This description provides information about the number of users involved in

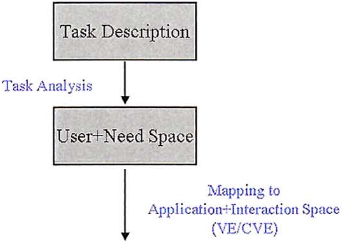


Figure 3.3: First level of the VE and CVE taxonomy graph.

the task, the type of material and the tools they use. It describes where the users stand and how they work together. This information is extracted following the *User's Task Analysis (UTA)*. This UTA determines the *User+Need Space (UNS)* which itself is the originator of the flow within the taxonomy graph. The UNS contains and groups the information extracted by the User Task Analysis of the User Task Description (UTD). It is recommendable 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 [47, 48]. From this thesis' point of view most of the Virtual Environments lack the addressing of user needs and thus result in a poor user satisfaction and usability [14, 50, 54].
Now it is possible to do an UTA of the UTD. The information extracted by the UTA facilitates the procedure of defining a User+Need Space. For the example described above the extracted information concerns:

- **Participants**

- **Objects**
- **Tools**
- **Objectives**
- **Constraints**
- **Actions**
- **Reasons for Cancellation**
- **Results**

With the help of this extracted information a User+Need Space (UNS) and its content is defined as described in the following sections.

3.3 Mapping from User+Need Space to VE and CVE

The User+Need Space is represented by an array-like representation format (see Figure 3.4).
This representation is more suitable for the purpose of this thesis since it provides an easier mapping between the requirements of the UNS and the features of the Virtual Environment under design. The first seven features denote representation components. In addition to the number of local and remote users their corresponding representations are included.
Although the UNS in Figure 3.4 is a UNS template, examples of different possibilities of realizations are added for clarification. When working two-handed, different input device combinations are shown, such as a combination of a stylus and a three button tool or the combination of a pinch glove and a cubic mouse (see section 3.4). These and other combinations are not obligatory, they are only 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 only of illustrating nature and show that more than one item could be taken under consideration. Thereby, if in the rows appears an enumeration, the first item or combination is interpreted as the most appropriate one. Then the application designer would 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. The next sections describe the items of the UNS in more detail.

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 3.4: The User+Need Space array-like representation.

3.4 Input/Output Device Combination and Work Mode

In this section the focus is on the design guidelines for combining input and output devices. It is obvious that not all six DOF input devices for interaction and output devices 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 for example the RWB as output device the user has the possibility of putting unused devices back on the table of the RWB. But of course this is not 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 factor for the selection of an adequate output device is the amount of users who work together at the same site as well as 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 might be a wall, a cylindrical projection or a CAVE rather than 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 in use. Thus input and output device combination is directly derivable from the User+Need space as all needs and requirements are already defined there. An example of determining the User+Need Space for the example of 3.2 is given in section 3.8.

The Work Mode is also determined by the users' tasks. According to this thesis' focus, the work mode is mainly determined by the user, sharing of data model and collaboration needed. In particular, different modes of work relevant to this thesis 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 of working alone where data sets are locally available or must be downloaded remotely, for instance from a simulation loop. No collaborative working is enabled at all in these two cases. The third and the fourth item describe collaborative working together using the same display system. Users are physically at the same place. The data sets are available locally again or have to be downloaded from a remote data server.
The last item is the more general one where at least two different sites work together. Now the shared data sets can either be provided by one or more members of the session or by an external data server. The work mode itself is important to determine the interaction metaphors described in sections 3.7.4 and 3.7.5.

3.5 Representation Components

Representation Components denote a very important part of Virtual Environments. They determine how the visual parts in the application are represented.

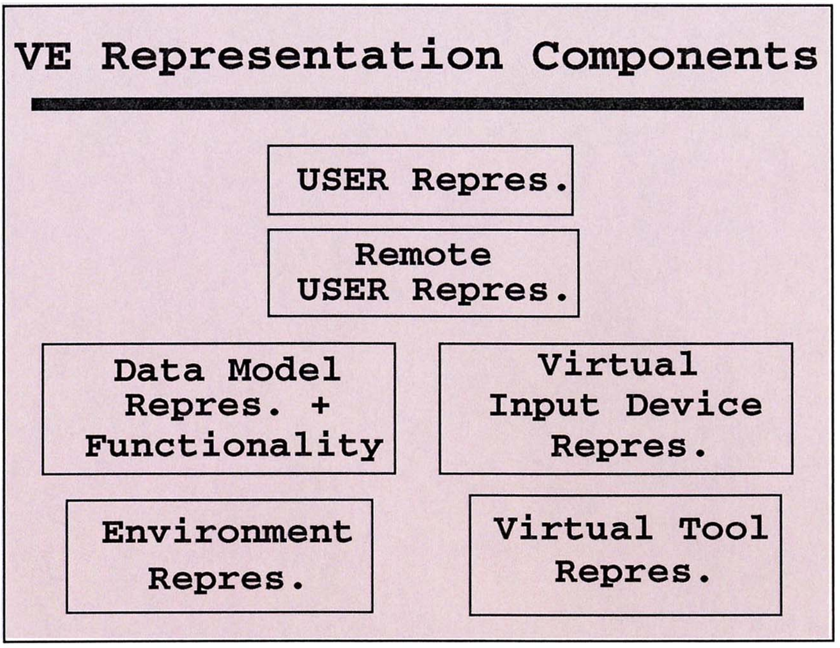


Figure 3.5: The VE representation components determine how the visual parts in the Virtual and Collaborative Virtual Environment are represented.

The components are (see Figure 3.5):

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

As shown in Figure 3.5 all components belong to a group although the *User Representation* has a special function. 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 for example in *Third Person Shooting* games.

The *remote user representation* represents the participating user or group of users from the other site. The aim of this representation form is to let the user or the group to appear present from a remote Virtual Environment. Therefore the factor of realism needs to be considered when designing an application. However, this depends on the task of the users. Sometimes more abstract user representations fit the requirements [80]. 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 representations that include realistic visual and physical models [23]. Research on using real-time video is using stereoscopic or mono video and different texture mapping and image manipulation techniques [67]. The advantages of video conferencing are the high realism and the ease in handling of the video texture in terms of positioning and scaling. The disadvantages are the transfer of video streams and network requirements. Also the alignment (matchmoving) of the texture with the virtual tool and input device representations needs to be considered.

The *environment representation* reflects the ambience of the users' physical environment. These representations can, for example, be an operation theater 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 users might 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 *data model representation* is the data set of interest. Depending on the application these data sets can be, for example, 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 chemist. Data sets of interest can either be abstract models or realistic synthetic models reconstructed from scanner data for example. The best representation format is determined by the possibilities of scientific visualization and the requirements of the user's task. When interacting with the data the different possibilities that denote its functionality have to be represented (see Figure 3.6).
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 needs to be represented in an easy to perceive way. There exist two main ways in VEs to show functionality to the user [47, 68]. 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 pro-

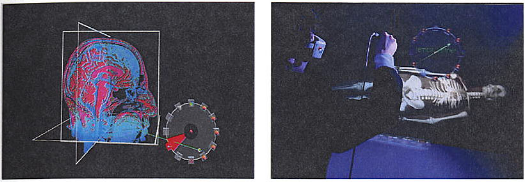


Figure 3.6: The ring menu belongs to one special data set. When a user asks the data set for its functionality this menu appears and shows the operations that are applicable to it. The menu is attached the user's hand position. It only follows the translation of the user's hand whereas the rotation of the user's wrist is used to intersect the "cake pieces" with the pick ray. Thus selection of operations is possible.

grammer have to find the most suitable way. Although a lot of work has been done in the area of HCI there is still a lack of guidelines for the specific set of applications which are the focus of this thesis. 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 (see Figure 3.6). It has been proven that it is a much better strategy to ask the data set for its functionality rather than to try and 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 [49, 71].

The *virtual input device representations* reflect the active physical input device the user has chosen. Examples of these representations are virtual colored rays when using the stylus or multiple button devices. These rays enable the user to see where the physical input device or the hand points to and they facilitate the selection process (see Figure 3.7).
The *virtual tool representations* reflect the active tool a user has chosen. These representations could be 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 (see Figure 3.7).

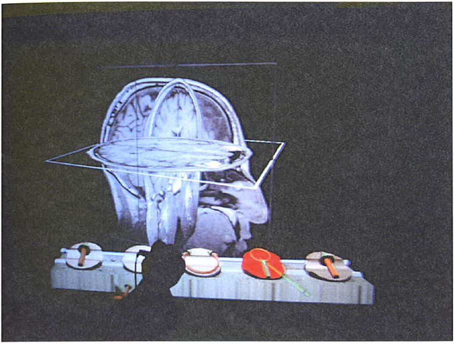


Figure 3.7: The toolbar does not belong to one special data set. It groups generic operations that are applicable to all kinds of data. The toolbar consists of buttons and virtual tool representations like 3D icons. When a button is pressed the icon is attached to the user's input device and replaces the virtual input device representation which is a pick ray in this example. The icon disappears from the toolbar highlighting that the tool is active.

3.6 Awareness-Action-Feedback Loops (AAF)

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 first the low-level makeup of interaction has to be understood. Therefore interaction tasks have to be narrowed down and interaction templates have to be found which can be combined to form more complex interactions.

Awareness-Action-Feedback loops denote such interaction templates (Figure 3.8). These AAF loops provide the possibility to understand and analyze very tiny steps in interactions.

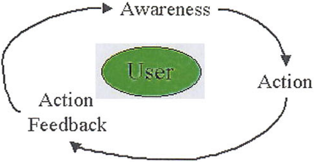


Figure 3.8: Awareness-Action-Feedback loops are interaction templates with which it is possible to analyse the low-level makeup of interaction.

3.6.1 Autonomous AAF Loop

Before explaining complex collaborative interactions it is started with autonomous interaction (see Figure 3.9). The autonomous AAF loop is divided

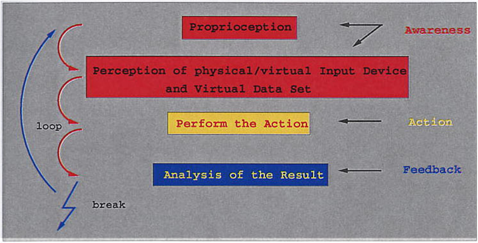


Figure 3.9: Diagram of the autonomous Awareness-Action-Feedback Loop.

into four blocks. The first two blocks belong to the awareness phase where the user starts with proprioception as defined by Boff et al. in [7]. The proprioception allows the user to become aware where it stands and looks to, the position and orientation of body parts like arms, hands and fingers and everything that is needed for interaction. It is the perception of the user 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 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 herself

the action phase follows. This action can simply be to move the hand holding the physical input device.
Upon completion of the action phase the feedback phase starts. This feedback is an action feedback without which it would not be possible to analyze the result of the action. In this case the user perceives the movement of the virtual tool representations as s/he moves the input device together with the hand. After the perception of the status of the situation the user decides 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 within the same loop.
The AAF loop is exemplified 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 and 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 the operation ? Is the nail positioned 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** until the task is completed.

3.6.2 Collaborative AAF Loop

Collaborative Awareness-Action-Feedback loops are of the same structure as the autonomous AAF loops (see Figure 3.10).
The main difference between them is that the collaborative AAF loop needs to additionally address collaborative requirements that are necessary when working in a team. Again the collaborative AAF loop starts with the proprioception

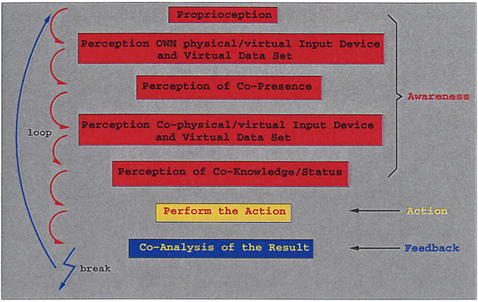


Figure 3.10: Diagram of the collaborative Awareness-Action-Feedback Loop.

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: Where is my partner, where does he look at, 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. It was possible to find out that knowing that your partner is aware of you is one of the most important steps in the awareness phase (see chapter 6). To know that your partner is aware of what you are intending to do and what you want to achieve with your action 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. The users can confirm this status check, by voice or with the help of a gesture like the "thumbs up". The action and the feedback phase are equal to the ones of the autonomous AAF loop.
In order to apply the collaborative AAF loop to a real scenario 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. Then it is possible to describe the whole interaction task from the carpenter's point of view who holds the

hammer.

1. **Proprioception** → *Awareness*
   Where am I ? Where do I look at ?
   Where are my hands, my fingers ?
   The same as before (see AAF loop).

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 ?
   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 my partner look at ?

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 at 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*
   Interaction of human body (hands, fingers etc.) and physical input device. Position the nail on the wood and position the hammer. The same as before (see AAF loop).

7. **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 the operation ? Is the nail positioned 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. We are not ready yet so proceed with pounding the nail.) The same as before (see AAF loop).

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

3.7 Operations, Metaphors, Interaction Techniques

Awareness-Action-Feedback loops like the ones shown in Figures 3.9 and 3.10 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 the 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 is explored. They can be data independent (i.e. basic operations such as selecting), or data dependent (i.e. slice through a 3D data volume). Refereing to the definition of interaction techniques and interaction tasks in section 2.1 of chapter 2 operations denote the action itself rather than the techniques that are used to put the operation into practice.
In this work three categories of operations are identified, generic operations, content specific operations and collaborative operations.

3.7.1 Generic Operations

Generic operations are used to manipulate virtual data sets of different kinds (see Figure 3.7). There exist a lot of generic operations and to mention only a few: *translation, rotation, zooming, dragging, pushing, deleting, grabbing, highlighting, selecting*.

In this approach selecting data is an even more fundamental generic operation since selection is the basis for all kinds of generic operations listed above.

3.7.2 Content Specific Operations

The other category of essential operations is content specific (see Figure 3.6). The virtual data sets determine additional operations that are meaningful depending on their features and nature. They are categorized as follows:

- change the mode of visualization
- change the appearance of the data (i.e. color and material attributes including textures, highlighting parts)

- change the geometric shape of the data (i.e. deformation, cutting, clipping, slicing)
- change the relationship of parts of the data (i.e. between different data sets or parts of the same)
- start/stop/pause of sequential data (i.e. videos, simulation loops, animations)
- reading and editing, descriptive text
- sonification of actions, events or text
- connecting/disconnecting with remote data servers or clients
- others

3.7.3 Collaborative Operations

All of the generic and content specific operations for the autonomous user mode can also be used in a collaborative session. Again data sets have to be translated, rotated and zoomed in or out. However, the operations need to be extended to include shared manipulation. Furthermore, additional operations are needed to establish and control a collaborative work session.

In the first category there are operations for:

- sharing of virtual data sets
- sharing different views on data sets
- sharing of operations

The data sets as well as the operations need to be distributed. A global operation box, similar to a tool-box, is the basis for sharing common operations in addition to the local operations at each site. Sharing different views of the data is important, for example when users like to concentrate on different aspects of the data set requiring different visualization modes.
In the second category there are operations for:

- establishing a session
- controlling positioning
- controlling conversation between participants

- terminating a session

More specifically, calling, hanging up, muting a video and audio connection at any time are generic operations dealing with the audio/video communication of remote users. Also switching between different remote partners or seeing and hearing them all at once is possible. Positioning of the remote participants' video representation in one's working environment allows control over the team's position and supports team dynamics.

3.7.4 Autonomous and Content Specific Metaphors

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). In this thesis three categories of metaphors are defined.
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 an interesting part of the data set, look closer, hear/touch interesting subparts, as well as additional ones like play video/TV, query information, can also be adapted from real-life paradigms.
However, there can be more than one way of combining operations to implement a metaphor. For example the teleport metaphor can make use of the zoom operation in order to scale the data set and let it appear larger to the user. In addition, it is also possible to apply a translation operation in order to either change the user's position to be closer to the teleporting point, or to move the data closer to the user. Depending on the application and the type of virtual data, one metaphor might be more intuitive than others. It is very useful to scale an object when observing interesting parts that retain the view on the surroundings. Moving closer to the object of interest could be useful for further operating on it. The metaphor that corresponds to Newton's law of action and reaction can either use the virtual data set or the user as point of reference. Concerning the effect, it makes no semantic difference whether the user moves around the object (*ego-centric manipulation*) or the object rotates around its axis and the user's point of view is stationary (*exo-centric manipulation*). Both implementations, ego-centric and exo-centric have their merits. The approach is to make the different metaphors transparent to the user and allow to choose the metaphor best suited for the task. This shows that generic, as well as content specific operations, can be used to implement metaphors.

3.7.5 Collaborative Metaphors

The *collaborative metaphors* are categorized in this work as follows:

- visual and verbal communication between users
- sharing viewpoint of participants
- virtual/verbal/tactile manipulation and sharing of data sets

Metaphors for visual and verbal communication include, working around a table, working next to each other, working on different parts of the data set, walkie-talkie and turn-taking verbal communication. The verbal communication metaphors, especially when using speech recognition, distinguish between voice commands and audio communication for talking to other participants. The user may want to give commands to the computer and to share them so that the remote site is aware of these commands. Using the walkie-talkie metaphor the remote site cannot disturb or interrupt the local user giving verbal commands to the system. Using the turn-taking metaphor the computer does not have to listen during the user's verbal communication. Important when using verbal communication is to simulate the real-life situation where the voice of a participant closer to the user is louder. This enhances the user's perception of presence and co-presence. Attaching the audio stream to the position of the video-avatar of the remote participant using localized sound sources is a possible implementation of this real-life paradigm.

Metaphors for sharing viewpoints include:

- sharing each other's viewpoint (look over the other's shoulder)
- same viewpoint (look through one's eyes)
- mirrored viewpoint (opposite side of table situation)

Finally, metaphors used for collaborative manipulation need to provide the possibility that all participating users manipulate a shared object at the same time using the same or different tools. This possibility is provided by the tug-of-war metaphor. An alternative metaphor that avoids deadlock situations is the tug-of-war with dead end metaphor. Each site receives two versions of the same shared object. One can be manipulated only by the local user and the other only by the remote user. This avoids conflicts but might require a bigger virtual space which can be limiting. Very basic locking mechanisms can also be used in order to avoid deadlock situations. Appropriate feedback types that indicate which user locks the data are required then.

3.7.6 Interaction Techniques

In contrast to the metaphors the interaction techniques determine how to support and implement the different types of operations. Again there are plenty of different ways to do this [49]. In order to make interaction in Virtual Environments richer and more intuitive, techniques have to be provided which use more than only one of our senses at the same time. Some interaction techniques that have proved to be quite adequate in terms of intuition as described in chapter 2 are:

- speech recognition
- tactile and force feedback
- menus
- virtual pick-ray
- toolbar/toolbox
- body-centered interaction
- gesture recognition
- olfactory interaction

Physical mnemonics and other senses have been successfully used to store and recall information relative to the body using hands, eyes, or even the whole body[71]. Depending on the available media and interaction devices, the defined operations can be implemented in different ways. For example, the selection operation can be implemented by recognizing simple voice commands or by the use of an interaction device (i.e. stylus) and a virtual pick ray. The perceived quality of interaction depends extremely on the interaction devices and their use. In order to name a few: *tracked shutter glasses, 6DOF pen-like stylus, multiple button tools with location sensors, data-gloves, the cubic mouse[41], tactile and force feedback devices, such as the Phantom from Sensable, and joysticks with location sensors, and many more described in chapter 2.*

The developer has to carefully select the most appropriate ones according to the VE. Recommendable techniques are those which enable the user to concentrate on the task and not on steering through menus and toolboxes.
Collaborative interaction techniques are the same as in the autonomous user mode. There is no need to develop new techniques in order to perform collaborative tasks. When being shared, menus, pick rays, voice recognition and other interaction techniques are used the same way as in the single user mode. A detailed taxonomy of interaction techniques can also be found in [12].

3.8 Example CVE design

In this last section the simulation of an example CVE design is made. With this example the reader will be able to understand the make-up and the process of CVE design making use of the VE/CVE design model developed from the taxonomy of influence factors at the beginning of this chapter.
It is possible to see how all representation components, work mode, operations, metaphors and interaction techniques can be chosen. For doing this the User Task description and the User Task analysis from section 3.2 is used. Thus the determination of a User+Need Space and the Application+Interaction Space has to be seen with respect to this UTD and UTA. Each item shows the corresponding representation briefly. The substantiation below each item indicates the choice for the special implementation.

User Representation

- no representation
  - reason: When using rear-projection systems no representation form for the user itself is necessary in contrast to using HMDs (see *output devices* and section 3.5 too)
- video image of the partner
  - reason: Using the video texture in rear-projection systems is a useful representation form for the remote partner (see section 3.5 too)
- Avatar
  - reason: Also a possible representation form. But as it is very important to see the movements of the partner working on the roof. So if using an avatar then it must include joints, muscles, motion etc. for the simulation of this task.

Representation and Functionality of the Data Sets

- virtual models of the wooden laths including the corresponding material properties of wood (ex. "woodiness")
- virtual models of the box with nails including the corresponding material properties of iron (ex. "steeliness" in order to pound them into wood)

Representation of the Environment

- virtual model of the roof (the viewpoints of the users have to follow gravity and to underly a collision detection with the roof)
- virtual models of the landscape, the sky, surrounding houses and other roofs which provide a feeling of height, depth and wideness (eventually acoustic or even physical wind representations)

Representation of the Tools

- virtual models of the hammers (including steeliness and weight if using force feedback systems)
- virtual models of the pair of pliers (see above)

Representation of the Input Devices

- virtual models of the users hands
  - note: Only useful when using data or pinch gloves as input devices
- virtual pick rays
  - note: Useful when using a stylus (pen-like input device)

Work Mode

- stand-alone, collaboratively and data sets are locally or remotely uploaded (for the case the two users are at the same site)
- distributed, collaboratively and data sets are provided by one of the sites, or by a remote (external) data server (for the case the two users are at different sites)

Input Devices - Output Device Combination

- stylus (practical)
  - reason : Easy to handle, suitable for both hands
- multiple button tool (practical)
  - reason : Easy to handle, suitable for both hands, due to the buttons more functional but here not really needed due to the simple task

- pinch or data glove (not very practical)
  - reason : Good representation of the using hand and thus direct manipulation is possible. But it can only be used with one hand. Further it is uncomfortable to use when trying to pass it to the partner unless the partner works remotely.
- video camera
- microphone (necessary when working distributed, collaboratively)
- Responsive Workbench (not practical)
  - reason : Handling the data set and working with the tools can be displayed. But working on a roof and having the collision detection with it is impossible to display within the limited view frustum.
- CAVE (practical)
  - reason : Handling the data set and working with the tools can be displayed. In addition to that working on a roof and having the collision detection with it is possible to display within the larger view frustum. Even when working stand-alone, collaboratively.
- Wall (not practical)
  - reason : As the display consists of one front screen and has no "floor" projection screen orthogonal operations are not possible without rotating the whole scene. This working metaphor does not exploit real-work-knowledge and thus is very unnatural (see section 3.7.4).
- cylindrical projection (not practical)
  - reason : The reasons are similar to the ones given for the wall display. One difference to the wall is that this display can provide a greater feeling of immersion through its surrounding character but the ground projection is also missing.
- speakers or headphones (practical and necessary when working distributed, collaboratively)

Generic Operations

- **select/grasp** - (either the nail or the hammer or the wood)

  Action

  * points with the virtual pick ray to the data set and presses a button of the stylus or multiple button tool
  * points with the finger to the data set and snips with the thumb and the middle finger when using pinch gloves

- **translate** - (e.g. in order to position the nail on the wood)

  Action

  * the user 'selects'/'grasps' the nail or the wood and moves the hand together with the selected object (this action is restricted because of the limited interaction space on top of the roof)
  * the user selects a special translate tool and applies this to the selected data set
  * the user selects a special drag tool which is a combination of a translate and rotate tool and applies this to the selected data set

- **rotate** - (e.g. in order to position the wooden laths to each other or on the roof)

  Action

  * the user 'selects'/'grasps' an end of the wooden laths and moves the hands (remember the laths are heavy → so two-handed interaction is required)
  * the user selects a special rotate tool and applies this to the data set
  * the user selects a special drag tool and applies this to the selected data set

- **zoom** - (e.g. in order to get a closer and better view on the data set)

  Action

  * the user 'selects' and 'translates' the data set and moves the hand closer to the eyes
  * the user changes its position (goes closer to the data set)
  * the user selects a special zoom tool and applies this to the data set

Content Specific Operations

- **change the geometric form of the data sets** - (e.g. when the nail goes inside the wood)

  Action

  * the user 'selects'/'grasps' the hammer, moves the hand and pounds on top of the nail

- **change of the view point** - (e.g. for verification purposes)

  Action

  * the user presses a button on the stylus or multiple button tool and 'rotates' the whole scene
  * the user presses a button on the stylus or multiple button tool and 'rotates' its viewpoint as long as the button is pressed

- **undo function** - (e.g. for pulling the nail out of the wood)

  Action

  * the user 'selects'/'grasps' a pair of pliers, selects the nail and pulls the nail while moving the hand
  * the user selects a special undo tool and applies this to the data set

Metaphors

- for the position of the partners:
  - *eye-to-eye contact, data set between the partners*
- for the communication between the partners:
  - *turn-taking conversation*
- for the rotation operation:
  - *rotate the data sets*
  - *walk around it but never rotate the whole scene as the users stand on the roof*
- for the zoom operation:
  - *change user's position (go closer to the object)*
  - *scale the data set itself*

- for the collaborative manipulation of the data sets:
  - *tug-of-war* : This metaphor allows both users to apply operations on the data set at the same time. No locking mechanism is implemented. The advantage is that the first user can hold the wooden laths with one hand and position the nail on top of them whereas the second user holds the wood with one hand and pounds the nail with the hammer in the other hand. The drawback is that both users exactly have to know what the other one is doing as uncoordinated interaction will result in a real tug-of-war situation.
  - *look through someone's eyes* : this metaphor enables the users to slip into the partners position and to have a look from the other side onto the data set. This metaphor is very useful when working opposite each other within a limited virtual work space.

Interaction Techniques

- tactile/force feedback : for the perception of the data sets and the tools specific weight, for tactile feedback when holding a hammer and a nail, force feedback when pounding nails
- menus : for the generic operations: zoom, rotate, translate
- virtual pick ray : for the selection with the stylus input device and in order to apply the operation
- sonification : for the acoustic feedback of the hammer blow
- Gesture recognition :

Types of Feedback

- changes of the data sets (highlight the nail if hit correctly, highlight the wood if the nail is positioned correctly, nail goes into the wood - wooden laths cannot be moved freely)
- acoustic feedback (hammer blow, 'hit/not hit'-Sound))

3.9 Conclusions

This chapter introduced a complex, theoretical interaction framework for Collaborative Virtual Environments. With the help of this framework Virtual

Environment designers and programmers are able to analyze user tasks and interaction cycles. The results of the analysis, then, determine a User+Need Space (UNS) which itself describes the appearance of the Collaborative Virtual Environment according to the visual representation. Performing the mapping of the UNS to the CVE, representation components are elaborated. Then together with the analysis of Awareness-Action-Feedback loops for autonomous and collaborative interaction cycles the desired Application+Interaction Space is determined.

The next chapter 4 determines an Application+Interaction Space for a predescribed two user task scenario. The focus there is on the use of two networked Responsive Workbenches. However, interface and application specific information concerning the coupling of a Responsive Workbench with a CAVE is also provided. In chapter 6 the developed applications are evaluated assessing usability and collaborative awareness.

Chapter 4

Application

This chapter describes an example Collaborative Virtual Environment in detail. It shows how all the issues mentioned in earlier sections of chapter 3 are used for selecting the most appropriate representation components, metaphors. operations and interaction techniques. In order to depict the User+Need Space as well as the Application+Interaction Space the same representations as in section 3.8 are used. The CVE presented in this chapter, with slight modifications is the basis for the description of the hardware setup in chapter 5 and the evaluation in chapter 6 assessing usability and collaborative awareness. However, the description of this decent CVE and its user scenario stands exemplarily for other rear projection based CVEs developed within this thesis.

4.1 CVE with two Responsive Workbenches

4.1.1 User Task Description

In order to perform a User Task Analysis (UTA) a detailed User Task description (UTD) needs to be provided. Due to its importance to the scientific community as well as to the novel ways education is carried out the following UTD describes a collaborative education scenario using an anatomical content.

Two users, an anatomy professor and a medical student work together on a virtual human data set. They stand opposite each other on either side of a table. They are able to walk around the table and to have a look from the other side onto the virtual human data set. The data set consists of human body 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, to drag and query extra information about them. The data set is used for anatomical education. Names of all bones can be queried as well as their *affiliation to the body's right or the left hand side. Additionally, test (exam) scenarios can be uploaded where different bones have to be compared with each other and inserted into the skeleton according to their functionality. The professor as well as the student have possibilities to verify the correct position of the bones inside the skeleton. In addition, both are equal in their possibilities while working on the data set.*

The User Task Analysis leads to the following application design according to all items that have been introduced in chapter 3.

4.1.2 Input Devices - Output Device Combination

- stylus (practical)
  - reason : Easy to handle, suitable for both hands, the professor can pass it over to another student if the session is extend to three or more users for example.
- three button tool (practical)
  - reason : Easy to handle, suitable for both hands. Due to the buttons more high-order functions available like changing the viewpoint, changing the test scenario etc.
- pinch or data glove (practical)
  - reason : Good representation of the using hand and thus direct manipulation is possible. But it can only be used with one hand. Further it is uncomfortable to use when trying to pass it to the partner. Within the described application they are not used, which is the first design decision.
- video camera as the remote partner representation is a video texture.
- microphone for communication.
- Responsive Workbench (practical)
  - reason : Handling the data set and working with the tools can be displayed excellently. For this kind of scenario the RWB is the optimal display system especially as the virtual human data fits within its viewing frustum almost without scaling.
- CAVE (not practical)

- reason : The CAVE is too oversized for the education session. Handling the data set does not require such a huge virtual interaction space as the CAVE offers. In addition using a CAVE at one or even both sites implies more rendering power.

- Wall (not practical)
  - reason : according to the fact that the data is needs to be viewed as on a table the Wall display system is not very practical since it does not provide this working metaphor.

- cylindrical projection (not practical)
  - reason : see argumentation for the CAVE

- headphones (practical), speakers (practical)
  - reason : The recommendation for using headphones instead of speakers is due to the reduction of acoustical feedback loops. As the scenario is designed to have one person on either site headphones are recommendable. However, headphones are not very comfortable to use, especially in combination with stereo glasses. A three or more user CVE with at least two users at the same site would need to have speakers in order to support natural communication at this site.

4.1.3 Generic Operations

- **select/grasp** - (either the bones or the whole body or the tools)

  Action

  * the user points with the virtual pick ray to the data set and presses a button of the stylus or three button tool
  * the user points with the finger to the data set and snips with the thumb and the middle finger when using pinch gloves

- **translate** - (e.g. in order to position a bone inside the skeleton or to obtain a better view of an object)

  Action

  * the user 'selects'/'grasps' the bone or the body and moves the hand together with the selected object (this action is restricted because of the limited interaction space in front of the RWB)

  * the user selects the group drag tool or the selective drag tool and applies it to either a bone or the whole body.

- **rotate** - (e.g. in order to position a bone inside the skeleton or to obtain a better view)

  Action

  * the user 'selects'/'grasps' the bone or the body and moves the hand together with the selected object (this action is restricted because of the limited interaction space in front of the RWB)

- **zoom** - (e.g. in order to get a closer and better view on the data set)

  Action

  * the user 'selects' and 'translates' the data set and moves the hand closer to the eyes
  * the user changes its position (goes closer to the data set)
  * the user selects the zoom tool and applies this to the data set

4.1.4 Content Specific Operations

- **change the geometric form when cutting the body skin**

  Action

  * the user calls the ring menu, 'selects' the knife icon through rotation of the wrist from the disc segment, gets a 3D frame attached to the left hand and moves the hand to cut the body skin.

- **change the visualization mode from solid into wireframe rendering**

  Action

  * the user calls the ring menu, 'selects' the 3D grid icon through rotation of the wrist from the disc segment that switches the body into wireframe mode and vice versa. The already selected tools remain in the user's hand during this operation.

- **change the visualization mode from opaque to transparent**

  Action

  * the user calls the ring menu, 'selects' the color lookup icon through rotating the wrist from the disc segment, gets a 3D slider, moves the slider with the selective drag tool attached to one of the input devices in the hands and thus changes the transparency.

- **change of the view point** - (for verification purposes and in order to get a better view.)

    Action

    * the user moves the head physically and looks from a different position.
    * the user presses a button on the stylus or three button tool and 'rotates' the viewpoint as long as the button is pressed.

- **query information** - (in order to query the name of the bone or to find out whether the bone belongs to the right or the left hand side of the skeleton.)

    Action

    * the user selects the special information tool from the toolbar through 'selecting' the 3D "i" letter icon, touches the desired bone and clicks the stylus button.

- **undo function** - (in order to remove the bones from wrong positions, to snap the bones into the destined positions or to verify the bone's position.)

    Action

    * the user selects the special snap back tool from the toolbar through 'selecting' the 3D hook icon, touches the desired bone and clicks the stylus button.

4.1.5 Metaphors

- for the position of the partners:
    - *eye-to-eye contact, data set between the partners*
- sharing viewpoint:
    - *look through the other's eyes or over the other's shoulder* : this metaphor enables the users to slip into the partners position and to have a look from the other side onto the data set. This metaphor is especially useful working opposite each other within a limited virtual work space or for teacher/student scenarios.
- for the communication and collaboration between the partners:
    - *ring-up and join session*
    - *turn-taking conversation*

- for the rotation operation:
  - *rotate the data set(s)*
  - *walk around the data set(s)*
- for the zoom operation:
  - *change user's position (go closer to the object)*
  - *scale the data set itself*
- for the collaborative manipulation of the data sets:
  - *tug-of-war* : This metaphor allows both users to apply operations on the data set at the same time. No locking mechanism is implemented. The advantage is that the first user can hold the skin cutter with one hand and position the bone inside the skeleton whereas the second user controls the body's position and size. The drawback is that both users have to know exactly what the other one is doing as uncoordinated interaction will result in a real tug-of-war situation.

4.1.6 Interaction Techniques

- menus : for the generic and content specific operations: zoom, rotate/translate, snap back, query, change geometric shape and visualization mode
- virtual pick ray : for the selection with the stylus input device and in order to apply the operation

4.1.7 Types of Feedback

- *highlight*: bones which are manipulated by the local user are highlighted in red, the ones manipulated by the remote partner are not highlighted. Respectively, for the remote user the bones manipulated by himself/herself are highlighted in red as well. The idea behind is that although a user needs to be aware of the remote partner's actions local own actions have higher priority.
- *pick ray and tool representation colors*: The tool pick rays and tool representations used locally are red and yellow for the ones of the remote user. Menu pick rays are represented locally in green and blue for the remote user.

- *textual feedback about remote viewpoint*: as the remote partner can change the viewpoint independently, the local user needs a feedback about this change. Therefore a text in the upper left corner of the RWB's back face reads "Student/Professor stands opposite you" or "Student/Professor stands beside you".
- *disappearing texture representation as local viewpoint feedback*: If the local user changes the viewpoint from eye-to-eye contact into the sharing viewpoint position the video representation of the remote partner disappears.

4.1.8 User Representation

- no representation
  - reason: When using rear-projection systems like the Responsive Workbench no representation form for the user itself is necessary in contrast to using HMDs.
- video image of the partner
  - reason: Using the video texture in rear-projection systems is a useful representation form for the remote partner. Additionally as the display systems are two Responsive Workbenches, the video image is mapped on the vertical face of the Responsive Workbench. This supports face-to-face communication and the perception of the remote user standing on the opposite side of the table.
- no Avatar
  - reason: Both users need to see each other like they would do in real exam situation. So the CVE programmers and designers need to create an extremely realistic avatar including muscles and body movements. As this requires much more additional computing and rendering power and network bandwidth a further design decision is formulated against using avatars.

4.1.9 Representation and Functionality of the Data Sets

- virtual models of the body skin, skeleton and heart.
  - the skin can be rendered in wireframe and in transparency mode.

- the heart does not need to beat since this is an anatomy session. It lies statically inside the body and only supports orientation. This implies no additional animation and rendering.
- the bones of the skeleton can be moved separately. Assigned to each bone is its name and the side of the skeleton it belongs to. This information can be queried by both users.

4.1.10 Representation of the Environment

- a virtual model of the environment is not recommended as the student user has to focus on the anatomical task. Although the aim of the virtual session is to transfer the learned to real world situations there is no need for rendering the inventory of a pathologic laboratory for example.

4.1.11 Representation of the Tools

- virtual models for the representation of the tools when they are in use:
  - group drag tool (3D cross with three orthogonal axes)
  - selective drag tool (pick ray with 3D arrow)
  - scale tool (magnifying glasses)
  - information tool (pick ray with 3D "i" letter)
  - snap back tool (pick ray with 3D hook)
  - skin cutting tool (3D frame)
  - transparency tool (3D slider)
  - wireframe tool (switch button)
- virtual models for the representation of the tool icons on the menus:
  - fixed 3D toolbar with 3D buttons and 3D tool icons
  - half transparent 3D ring menu disk consisting of 12 disc ("cake") segments
  - group drag tool (3D three cross with two axes on the fixed 3D toolbar button)
  - selective drag tool (3D arrow on the fixed 3D toolbar button)
  - scale tool (magnifying glasses on the fixed 3D toolbar button)
  - information tool (3D "i" letter on the fixed 3D toolbar button)
  - snap back tool (3D hook on the fixed 3D toolbar button)

- skin cutting tool (3D knife on the ring menu disc segment)
- transparency tool (3D colored lookup icon on the ring menu disc segment)
- wireframe tool (3D grid button on the ring menu disc segment)

4.1.12 Representation of the Input Devices

- virtual pick rays
  - reason: useful when using a stylus and three button tool. Some of the tool representations use pick rays in addition to their 3D icon (see above). The functionality of the pick rays is distinguished by their color. So, the pick ray for calling the ring menu is green (blue for the remote user) whereas pick rays connected to tools are red (yellow for the remote user).

4.1.13 Work Mode

- distributed, collaboratively and data sets are provided by one of the sites, or by a remote (external) data server.

The representation components as well as the operations, metaphors and interaction techniques are put into practise designing and implementing a Collaborative Virtual Environment for education purposes. Two snapshots of a real session in this CVE are shown in Figure 4.1.
The images show the two users working together (collaboratively) on an anatomical virtual human data set. Each snapshot is made from either site of the collaborative scenario. In the upper left and right corner the information about the partners viewpoint is displayed. This provides the feedback necessary for communication purposes. In the middle of the back face of each RWB the small finger sized cameras are visible. They are mounted this way to minimize the viewing angle onto the remote partner. If the camera would have been placed outside the viewing frustum, algorithms for image reconstruction of the front view were necessary. With these CPU intensive reconstructions the real-time requirement of CVEs would not be fulfilled.

4.2 CVE design using other display systems

4.2.1 CAVE-RWB

Many more applications using different display combinations are implemented within this thesis. The same application provided above is implemented using

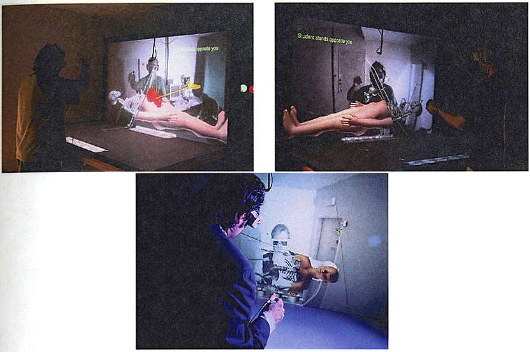


Figure 4.1: Snap shot taken from real sessions. For taking photos, the scene is rendered in monoscopic view. The upper left photo is taken from the remote RWB site than the upper right image.

a RWB-CAVE configuration. Images from the real application are shown in Figure 4.2.
Although the application is designed after the User Task Analysis of the User Task Description provided in section 4.1.1 some representation components are different when using a RWB-CAVE display combination instead of two RWBs. For increasing readability these differences are presented in the following rather than providing the whole User+Need Space and Application+Interaction Space again. Components that are not presented here do not change in the new display system combination.

Input Devices

- stylus (practical for RWB and CAVE)
  - reason : Easy to handle, suitable for both hands, the professor can pass it over to another student if the session is extend to three users or more.
- three button tool (practical for RWB and CAVE)

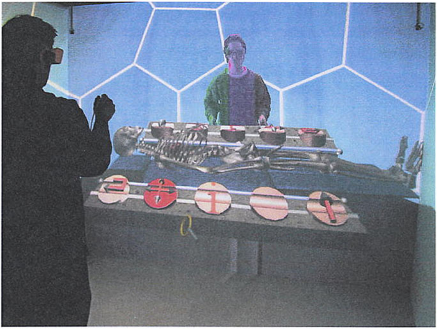


Figure 4.2: The snapshot shows two users collaborating in the CAVE. The desk with the data model represents the reference frame for the interacting user in the CAVE as the remote user works at the smaller display system.

  - reason : Easy to handle, suitable for both hands and usable in combination with both display systems. Due to the buttons more high-order functions are available like changing the viewpoint, changing the test scenario etc.. This is the reason why the three button tool instead of the stylus is used in the CAVE.

- joystick (practical for CAVE)
  - reason : As the CAVE display offers a much bigger viewing frustum and thus has a need for navigation, the users are provided with a special wand (joystick) for changing the viewpoint in this environment. The joystick is not usable at all in combination with a RWB. Here it is much better to ensure the model fits exactly into the CAVE boundaries and then walk within the display system environment.

Representation of the Environment

- In the CAVE display system a Virtual Environment representation is added to the data set and its functionality. This representations include inventory, surroundings, floor etc. The reason is simply that the viewing frustum of the CAVE of approximately $27m^3$ has to be filled in a way that the application does not seem to soar in a black and empty space. This means that the environmental representation is not used in order to facilitate applying the learned in real world but rather to provide a reference frame for the application.

Metaphors

- for the handling of the toolbar grouping the generic operations:
  - *vendor's tray metaphor*: This metaphor is used for positioning the toolbar dynamically. This means that the CAVE user is carrying the toolbar like a vendor's tray. On the disposal of the RWB the toolbar grouping the generic operations is fixed. The handling of the ring menu grouping the content specific operations remains the same.

Types of Feedback

- *Representation of the smaller display system within the bigger*: Combining the RWB with the CAVE implies problems that might occur due to the different sizes of the viewing frustums. So it might occur that the CAVE user translates the data set within the bigger viewing frustum not knowing that the data set is clipped because it is outside the viewing frustum of the user at the RWB. This means that the user of the bigger display system needs to perceive the size of viewing frustum of the smaller display as a feedback. Here this problem is solved in providing a model of the smaller display system (Responsive Workbench) within the CVE of the bigger display system (CAVE). Therefore the RWB is represented by a normal table not showing the back face of the real RWB.

4.2.2 CAVE-RWB-Cylindrical Display

The last application example is a CVE for a combination of a Responsive Workbench, a CAVE and a cylindrical display (Cone) like the one shown in Figure 1.5. The idea behind this application is to provide three teams of players with tools to complete different tasks in an adventure game CVE. Depending on the display systems the teams have different abilities and functionalities within the game. The aim of the adventure game is to find three magic stones and to place them in a magic stone circle in order to discover the remaining secret. Thereby the teams are not playing against but with each other and the game's stones can only be found and placed when they are working together being aware of the co-player's abilities and position. Figure 4.3 shows some preliminary results as this game is still under construction.

4.3 Conclusions

This chapter described the CVE application details for the combination of two RWBs and a RWB with a CAVE. Starting with a precise User Task Description (UTD) all necessary representation components, metaphors, operations and interaction techniques are determined. With this it is possible to create the User+Need as well as the Application+Interaction Space for the application. The remainder of this chapter provided information about a currently developed CVE combining a Responsive Workbench, a CAVE and a Cylindrical Display (Cone). The UTD for this application describes an adventure game based multi-user experience for practising team work. In chapter 5, it is shown how the former CVE application is implemented. It is evaluated according to usability and collaborative awareness assessments in chapter 6.

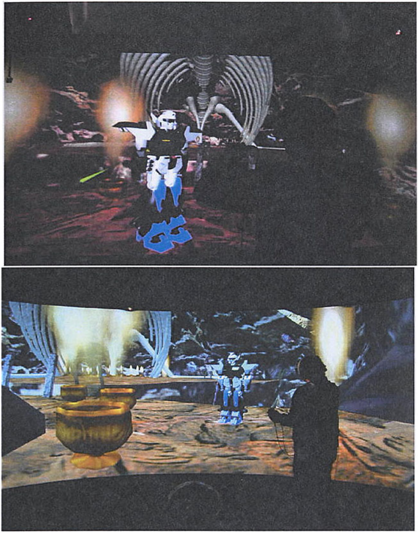


Figure 4.3: The images show some preliminary results of the collaborative game which is still under construction. The upper one shows the user interacting in GMD's cave-like CyberStage whereas in the lower one the user interacts in the CVE using the cylindrical display as output device.

Chapter 5

Implementation

This chapter provides a technical description of the implementation of two scenarios, one using two distributed Responsive Workbenches and the other one using a distributed Responsive Workbench and a CAVE. The first section describes the setup with respect to the hardware configuration used, introducing the rendering and interaction equipment. The second section talks about the software configuration describing the audio/video conferencing as well as rendering and distribution. Code fragments show the application programming briefly and flow charts are used to represent the combination of different techniques, operations, data and equipment.

5.1 Hardware Configuration

Most projection-based VEs and CVEs show almost the same hardware configuration. In the first place there is a computer that processes data and renders it to the screen as well as a tracking device that measures the position and orientation of the user's viewpoint. This tracking data is read by the rendering machine in order to determine the correct perspective view onto the virtual scene from the user's viewpoint. The hardware configuration implemented in this thesis includes input devices for interaction, computers for rendering and distribution and computers for video and audio streaming. Additionally it includes equipment like shutter glasses, infra-red emitters, cameras as well as microphones and headphones.

5.1.1 RWB-RWB configuration

For the distributed RWB-RWB setup the hardware configuration shown in Figure 5.1 is used [47, 51]. For the rendering two SGI ONYX workstations are used with at least one Infinite Reality (IR2) graphics pipe each. The reason is

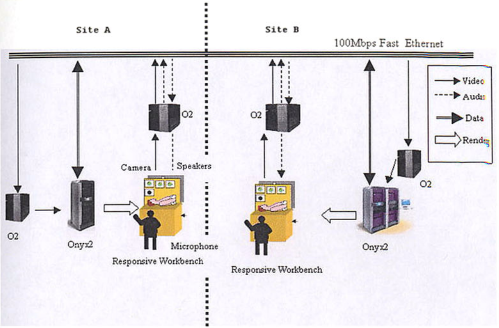


Figure 5.1: Schematic of the implemented setup with two Collaborative Responsive Workbenches.

that the SGI pipe architecture consists of two channels that can each render a stereo image of a typical resolution of 1280x1024 pixels at either 96 Hz or 120 Hz stereo. Additionally a minimum of four MIPS R10000 processors are needed. Obviously, a lower cost hardware can be used both for rendering and for the audio/video communication. The availability of the particular equipment and its high performance provide an easy choice.
For the tracking of the user's head and input devices the Fastrak tracking system from Polhemus is used. Therefore a Polhemus sensor is attached to the left earpiece of the Crystal Eyes shutter glasses. For interacting within the Virtual Environment a Polhemus stylus and an own custom-made three button tool are provided. The tracking sensors are attached to the input device too, as the computer needs to know where the user holds the input devices and points them to. The sensors' position and orientation are measured electronically by a receiver attached to the front side of the Responsive Workbench. Through calibration and transformation the co-ordinate system of the tracking system is matched with the world co-ordinate system of the CVE.
For communicating with the remote partner wireless microphones and headphones are used. The video and the audio conferencing is handled by two O2

workstations. The decision for using these type of computers was mainly influenced by the availability of the O2 MVP video cards. These special video cards are equipped with a special motion jpeg encoding chip. The video streams are grabbed directly from the infra-red video camera. The infra-red cameras are necessary since the light in the laboratory is dimmed in order to perceive the rendered images with high contrast and brightness. After the video is received by the O2 workstation the stream is transferred to the DIVO video boards of each ONYX workstation. The same O2 that manages the video conferencing also manages the audio connection. The audio stream is grabbed from the wireless microphones and then send to the other O2 where the headphones are plugged in. The reason for using headphones is to avoid acoustic feedback loops which disturb the communication between the remote partners. As soon as more than one user is working at the same site more headphones or normal speakers have to be taken in order to provide the other person(s) with audio perception too.

5.1.2 RWB-CAVE configuration

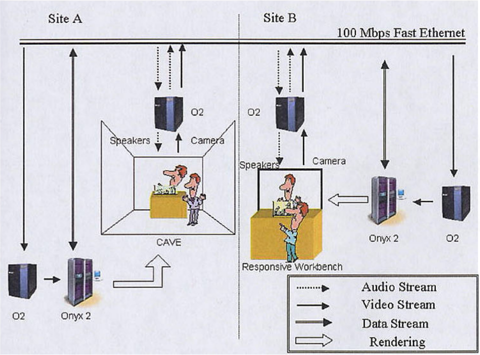


Figure 5.2: Schematic of the implemented setup with one Responsive Workbench and the GMD's cave-like CyberStage.

The hardware configuration for the distributed RWB-CAVE setup is sim-

ilar to the described RWB-RWB setup. Again two SGI ONYX workstations are used. The same computer is used for driving the RWB as in the other setup. For rendering four stereoscopic images (eight images in total) on the four walls of the CAVE, an ONYX workstation with at least two graphics pipes is needed. In the used configuration two or four Infinite Reality (IR2) graphics pipes are used. Each machine needs at least four MIPS R10000 processors. The performer application, drawing and culling processes are each running on one processor and the process running on the forth processor is importing the tracking data to the scene graph. As the application needs video conferencing facilities and tracking of two input devices in addition to the tracking of the viewpoint and the rendering, six MIPS R12000 processors are used to ensure a rendering frame rate of approximately $30Hz$.
As the video representation of the remote partner in the CAVE is offered without camera background additional video hardware is necessary (see Figure 5.3). Therefore special hardware chroma keyers are used for segmentation in order to determine the regions in the video images where the participant appears. The video streams sent and received by the O2 workstations are transferred to an Ultimate hardware keying device. After the remote partner has been cut out of the surroundings through filling the subtracted regions with transparency values the remaining video stream is transferred to the DIVO video boards of the ONYX workstation. For a clean and precise chroma keying homogeneous lighting is essential. This is contradictory to the light in the laboratories which is dimmed for perceiving the rendered images with high contrast and brightness. Additionally, the infra-red cameras produce black/white images only. However, if the user is wearing bright clothes and the keying color is black it is possible to subtract the user from its background. Especially helpful is the design of a ring which consists of five to ten infra-red LEDs. This ring attached around the infra-red camera makes it possible to illuminate the user directly as the user is supposed to look into the camera. When using infra-red light it should be ensured that the frequency does not disturb the infra-red driven shuttering of the stereo glasses.

5.2 Software Configuration

The software used for the implementation of the CVE is mainly Avango. The basic concepts of Avango are described already in section 1.3.3.
Several attempts to offer toolkits for distributed VE application development have been made recently (eg. VR Juggler [5], DIVE [24], WTK [84, 85]). These toolkits provide various degrees of support for network based communication between the distributed processes that form an application. However, using

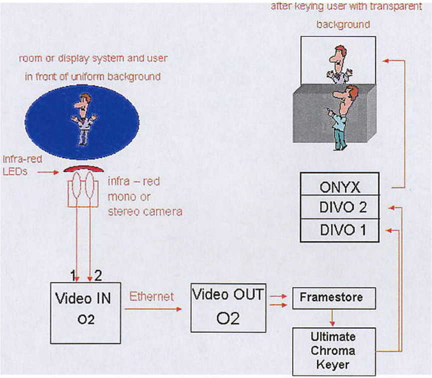


Figure 5.3: The video streams sent and received by the O2 workstations are transferred to an Ultimate hardware keying device. After the remote partner has been cut of out the surroundings through filling the subtracted regions with transparency values the remaining video stream is transferred to the DIVO video boards of the ONYX workstation.

these facilities often requires significant effort from the programmer. Normally, only parts of the application state are shared between the distributed processes such as transformation matrices which describe object positions. Sometimes explicit specification of communication endpoints for shared object attributes is necessary. The resulting database duplication problem, ensuring that all processes work on consistent copies of the shared database, is left as a challenge to the application programmer. This can be a tedious and error prone task especially when additional processes join an already running distributed application. Avango provides programmers with the concept of a shared scene graph, accessible from all processes forming a distributed application.

The software that handles the audio/video conferencing, however, is designed to run independently from Avango. Before implementing conferencing software, already existing audio/video conferencing toolkits were considered for integration. The requirement of having PAL or at least NTSC sized video resolution at 20-30 fps is the reason why toolkits like Mircosoft's netmeeting, public domain MBone tools or the well known public domain Vic and Rat tools cannot be integrated in the CVE. These toolkits are designed for low bandwidth connections such as modem, ISDN or DSL links, transferring icon-sized video conference images. Although some tools like the H263 encoding algorithm from Telenor are able to compress and send 4CIF (704x576 Pixels) sized video streams, but the frame rate drops to almost 0.5-1.0 fps. The developed audio/video conferencing software within this thesis is able to send PAL video streams over the network at 25fps and to integrate them into a CVE in real-time. Therefore the software handles bi-directional audio/video connections.

The first part of the following section describes the Avango field interface and how scene graphs are distributed. Additionally it provides code fragments to give an idea about the programming effort for implementing CVE applications.
The second part deals with the software that enables the audio/video conferencing in mono and stereo. In addition, modifications are described which are necessary for integrating stereo video textures into a VE using Avango.

5.2.1 Distributed Scene Graphs with Avango

Avango 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 [96]. The Avango framework is based on Performer to achieve the maximum possible performance for VE and CVE applications. Performer provides a C++ scene graph API which allows for flexible representation of complex geometry [87, 88, 89]. Advanced rendering tasks like culling, level-of-detail switching and communication with the graphics hardware are all handled by Performer. Whenever the underlying hardware allows, Performer utilizes multiple processors and multiple graphics pipelines. The toolkit also provides a collection of geometry loaders for the most popular as well as some uncommon file formats.

Fields and Fieldcontainers

The efficient implementation of a generic scripting and streaming interface for heterogenous objects requires additional *meta* information about object attributes and their types, and a way to access those attributes without knowing the exact type of the containing object. The C++ programming language does not treat classes as first class objects, so that this information is not easily available on a language level. Performer uses a member function API to access the state attributes of an object. A symmetric pair of get and set functions exists for each attribute. Setting one attribute may change another attribute of that object as a side effect. However, no abstract information about the number of attributes, their type, and their value can be obtained from an object via the Performer API. To overcome this problem Avango introduces the notion of *fields* as containers for object state attributes to Performer objects. Fields encapsulate basic data types and provide a generic interface for script-

```
template<class T> class fpSingleField : public fpField {

public:
  virtual fpType   getTypeId() const;
  virtual void     setValue(const T& value);
  virtual const T& getValue() const;
  virtual void     setSchemeValue(Scheme value);
  virtual Scheme   getSchemeValue() const;
};

template<class T> ostream& operator<<(ostream& stream,
                      const fpSingleField<T>& field);

template<class T> istream& operator>>(istream& stream,
                      fpSingleField<T>& field);
```

Figure 5.4: Part of the `fpSingleField` interface. `fpSingleField` is a template class and is used to instantiate single value fields on basic data types.

ing and streaming. They are implemented as public class member functions and are thus inherited by derived classes. They are directly accessible by client classes and are Avango's premier interface to object state attributes. There exist four different types of fields. *Single-fields* contain one basic data type value, whereas *multi-fields* contain a vector of values. A multi field can contain an arbitrary number of values. To bridge the Performer method based API to the Avango field oriented API, *adaptor fields* are used. They will forward `getValue()` and `setValue()` requests to the appropriate get and set functions of the Performer API. This ensures that Performer related state information is correctly updated according to field value changes, and possible side effects are properly evaluated.
The implementation of fields makes use of C++ templates which allow to parameterize the `fpField` class with the required data type. As an example for

the field class API, part of the template class definition for the single field is shown in Figure 5.4. The `getTypeId()` method provides access to the run-time type information for a field. The returned type identifier is used for run-time type checking and to provide type information for field values which are written to streams.
Access to the field value is provided by the `getValue()` and `setValue()` methods. Alternatively, a scheme representation of the field values can be provided or obtained via the `getSchemeValue()` and `setSchemeValue()` methods. For each field class a pair of stream operators exist which allow serialization of the field value into a stream, and the reconstruction of the field value from a stream.
Avango provides a *fieldcontainer* interface which represents the state of an object as a collection of fields. A fieldcontainer can be queried for its number of fields, and any of the fields themselves. Relevant parts of the fieldcontainer interface are shown in Figure 5.5. This, together with the generic scripting and

```
class fpFieldContainer {

public:
  int              getNumFields();
  fpFieldPtrVec&   getFields();
  friend ostream&  operator<<(ostream& stream,
                              fpLink<fpFieldContainer> fc);
  friend istream&  operator>>(istream& stream,
                              fpLink<fpFieldContainer> fc);

protected:
  virtual void notify(fpField& field);
  virtual void evaluate();
};
```

Figure 5.5: `fpFieldContainer` encapsulates the state information of an object and represents it as a collection of fields.

streaming interface of fields, allows to provide complete scripting and streaming functionality at the fieldcontainer level without knowing the exact type of the underlying object. This extents, at no extra cost, to all classes further derived from `fpFieldContainer`.

Fieldconnections

Much like SGI's Open Inventor Toolkit, Avango introduces the concept of *fieldconnections* [102]. Fields of compatible type can be connected in a way that whenever the value of the *source field* changes, it is immediately forwarded to the *destination field*. Using fieldconnections, a data-flow graph can be constructed which is conceptually orthogonal to the scene graph. Avango utilizes this mechanism to specify additional relationships between nodes which cannot be expressed in terms of the standard scene graph. This allows for the easy implementation of interactive behaviour and the import of real world data into the scene graph. The evaluation of the data-flow graph is performed once per rendering frame. Fieldconnections forward value changes immediately, so that there is no propagation delay for cascaded connection paths in the graph (see Figure 1.9). Loops are detected and properly resolved. A fieldcontainer can implement side effects on field changes by overloading the `notify()` and `evaluate()` member functions. Whenever a field is set to a new value, the `notify()` method is called on the fieldcontainer with a reference to the changed field as an argument. The fieldcontainer can do whatever is necessary to achieve the desired effect, including the manipulation of other fields. After all notifications for all fields on all fieldcontainers have been made for one frame, the `evaluate()` method is called on each fieldcontainer which had at least one of its fields notified. This allows the fieldcontainer to perform actions which depend on more than one updated field value for each frame.

Nodes

Fieldcontainer adaptions exist for all Performer node classes (`pfGroup`, `pfDCS`, `pfLOD`, etc.) and most of the Performer object classes (`pfGeoState`, `pfMaterial`, `pfTexture`, etc.). By convention the Avango nodes exchange the Performer `pf` prefix with the `fp` prefix. The Avango scene graph is built out of nodes as shown in Figure 1.9. The possibility to mix Avango nodes with Performer nodes in the scene graph can be conveniently used to define new nodes with interesting functionality. The `fpFile` node, for example, is derived from the adaption node `fpDCS`. It inherits the interface of `fpDCS` which consist of a `Children` and a `Matrix` field. The `fpFile` node adds a `URL` field of type `string`. With an overloaded `notify()` method, `fpFile` reacts to changes of the `URL` field by retrieving a geometry from the specified URL and adding it to its list of children. Thus, `fpFile` imports the geometry into the scene graph, and can be seen as an opaque handle to it. Subsequent changes to `URL` will result in replacement of the old geometry with the newly specified geometry.

Sensors

Sensors represent Avango's interface to the real world (see Figure 1.9). They are derived from the `fpFieldContainer` class but not from any Performer node, and thus cannot be inserted into the scene graph. Sensors encapsulate

the code necessary to access input devices of various kinds. The data generated by a device are mapped to the fields of the sensor. Whenever a device generates new data values, the fields of the corresponding sensor are updated accordingly.
Fieldconnections from sensor fields to node fields in the scene graph are used to incorporate device data into an application. The `fpTrackerSensor` class provides an interface to six degree of freedom trackers like the Polhemus Fastrak devices (see Figure 5.6). Avango utilizes a device daemon process which

```
class fpTrackerSensor : public fpDeviceSensor{

public:
  fpSingleField<string>   Station; // inherited
  fpSingleField<fpMatrix> Transform;
  fpSingleField<bool>     Button;

};
```

Figure 5.6: The Avango sensor classes map data values from external devices to fields.

handles the direct interaction with the devices via serial line or network connections. The daemon updates the device data values into a shared memory segment, where the `fpDeviceSensor` classes can access them. A *station* name is used to identify the desired device data in the shared memory segment, and every `fpDeviceSensor` class specifies this identifier in its `Station` field. After connecting to the device daemon, the `fpTrackerSensor` class provides the current position and orientation information from the selected tracking device represented in form of a matrix in its `Transform` field. By connecting the `Matrix` field of a `fpDCS` node to the `Transform` field, the subtree rooted by the `fpDCS` node will move according to the tracker movement reported from the selected station.

Scripting

The development of Virtual Environment applications often follows a highly iterative approach. Many VE toolkits and frameworks do not account for this situation as changes and reconfigurations require recoding in C or C++ and recompilation of parts or even the whole application. An interpreted scripting language which has a binding to all relevant high-level object interfaces in a framework can greatly reduce the burden on the application programmer and

```
;; instantiate a fpFile node and attach it as child to the
;; scene-root node
(define geom (make-instance-by-name "fpFile"))
(-> (-> scene-root 'Children) 'add-1value geom)

;; load a geometry from the following path
(-> (-> geom 'Filename) 'set-value "./graphics/iv/jaw_bone.iv")

;; instantiate a Drag Tool Node and activate it
(define drag-tool (make-instance-by-name "fpDragTool"))
(-> (-> drag-tool 'TimeIn) 'connect-from (-> time-sensor 'Time))

;; instantiate a dragger - a special matrix dragger
(define geom-dragger (make-instance-by-name "fpMatrixDragger"))

;; assign the dragger to the geometry
(-> (-> geom 'Dragger) 'add-1value geom-dragger)

;; connect the dragger's matrix with the geometry's matrix
(-> (-> geom 'Matrix) 'connect-from (-> geom-dragger 'Matrix))
```

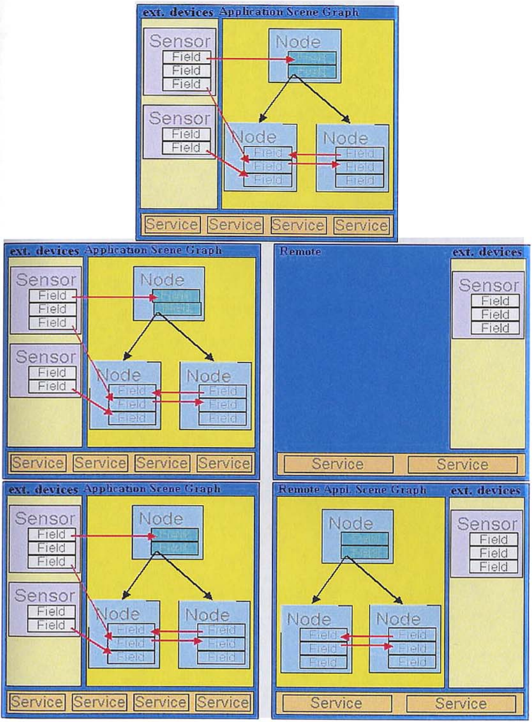
Figure 5.7: A `fpFile` node is instantiated and loads an Inventor file from the path specified in the *'Filename* field. By making the file node a child of the *scene-root* the associated geometry gets rendered. After instantiating a `DragTool` and configuring it, the `DragTool` checks for intersections between the input device representation geometry and all other geometry in the scene. If an intersection with *geom* is detected, a special mechanism looks for an instance of a dragger being assigned to *geom*. In this case the matrix field connection between *geom* and *geom-dragger* is executed and the jaw bone geometry follows the movements of the input device.

will significantly shorten the development cycle. No recompilation is necessary and modifications can be applied to running applications. As already described in section 1.3.3 Avango features a complete language binding to the interpreted language Scheme. It uses the *ELK* Scheme implementation which is a small and elegant Scheme interpreter and is especially suited to serve as an extension language for C and C++ programs. The Avango scheme binding is based on the field and fieldcontainer API's of the Avango objects. For all basic data types that are used to instantiate field classes a scheme representation with all necessary access functions exists. The basic data types are passed by value to and from the Scheme interpreter and can be handled like any other built-in Scheme type.
Avango objects such as nodes and sensors are handled by reference. This is

implemented by providing a binding for the `fpLink` class. `fpLink` values are again passed by value to the Scheme interpreter so that references to Avango objects are properly reference counted. Scheme uses a garbage collector to reclaim memory from objects which can no longer be accessed by the interpreter. When an `fpLink` value is garbage collected, the reference count on the associated Avango object is decremented accordingly.
Avango objects can be created from Scheme by providing the name of the desired object class as an argument to the `(make-instance-by-name class)` function. The object is instantiated, and a reference is handed back in the form of a `fpLink` value.
A set of access functions allows access to the fieldcontainer and the field interfaces of Avango objects. Figure 5.7 shows an example script which instantiates an `fpFile` node and connects it to an instance of a `fpMatrixDragger`. A `fpMatrixDragger` contains the position and orientation of the input device. With a field connection between this dragger and a geometry the latter one follows the movements of the input device. Because ELK Scheme is an interpreted language, every Avango application can provide a command-line interface, where Scheme commands can be entered at run-time. All Avango objects which are defined by a scheme script can be accessed and manipulated while the application is running.
The Avango scripting interface suggests a two phased approach for the application development. Complex and performance critical functionality is implemented in C++ by subclassing and extending already existing Avango classes. The application itself is then a collection of Scheme scripts which instantiate the desired Avango objects, set their field values and define relationships between them through field connections. The scripts can be written and tested while the application is running. This leads to very short turnaround times in the development cycle and provides a very powerful environment for rapid application prototyping.

Distribution

As already described Avango objects are fieldcontainers that encapsulate the object state in a set of fields. The streaming interface of the field and fieldcontainer classes allow for a very elegant implementation of the distributed object semantics.
Distributed object creation in Avango is a two stage process. First a local object is created which is then, in a second step, migrated to the desired distribution group (see Figure 5.8). The migration involves the announcement of a new object to the distribution group and the dissemination of the current object state to all group members. For this the streaming interface of the


ext. devices
Application Scene Graph
Sensor
Field
Node
Service
Remote
Remote Appl. Scene Graph

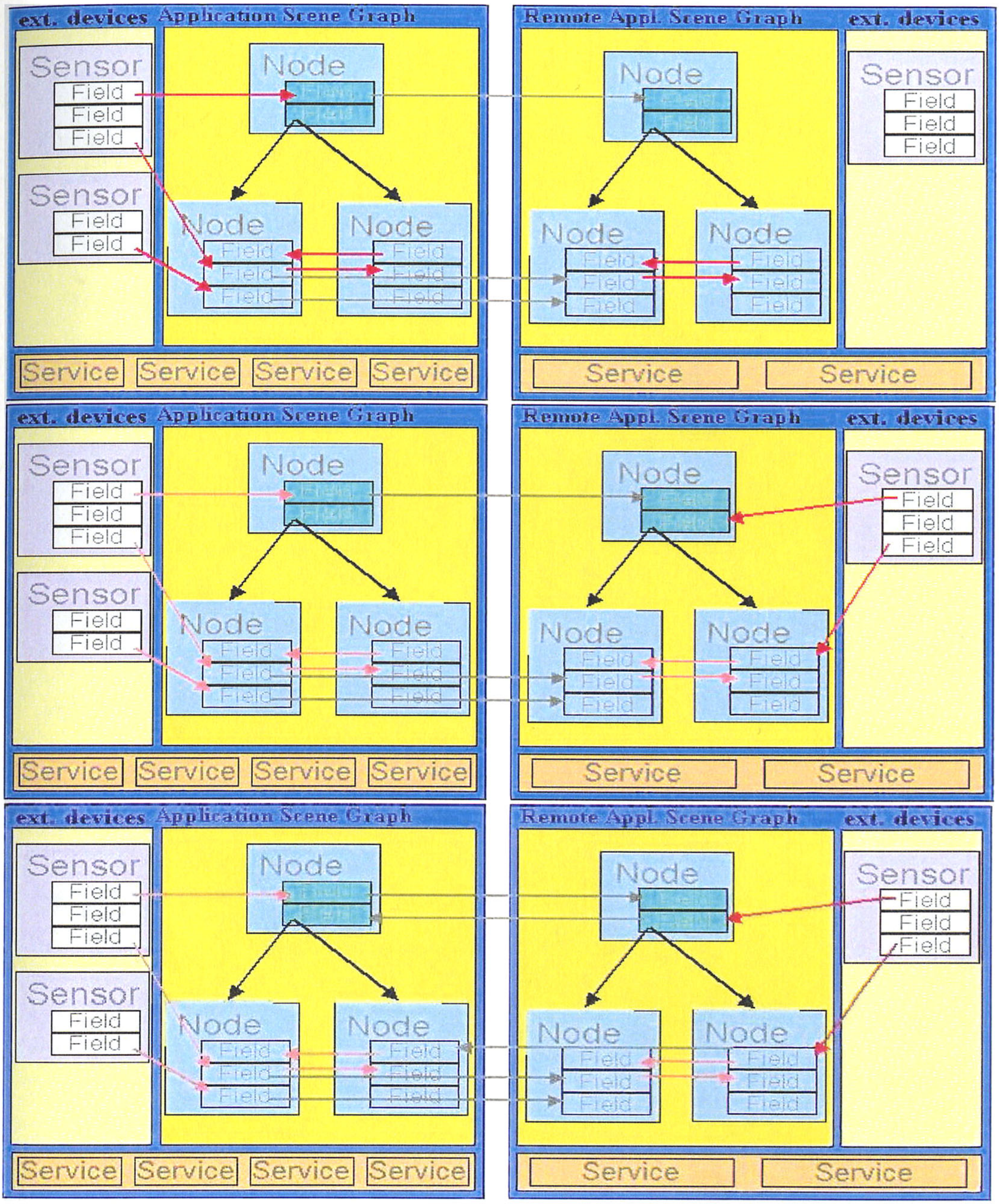


Figure 5.8: Illustration of a scene graph distribution using Avango. The six steps show how the scene graph is copied to a remote site and field values are updated. The last two steps illustrate how the participants who where joining the group at a later time are able to interact on the received data and propagate changes back to the first user.

fieldcontainer is used to serialize the object state into a network data buffer which is then sent to all group members. The group members will reverse this process and create a copy of the object from the serialized state information. The newly created distributed object will now exist as a local copy in each of the participating processes.
In Avango object state is accessible in terms of object field values. Whenever a field value on a distributed object is locally changed the new value is streamed to a network buffer and sent to all members of the distribution group in order to keep the distributed copies of that object synchronized.
The parent child relationship between objects in the scene graph is represented by a multi-field of reference values on the parent node. Because all field values including `fpLink<>` types are streamable, the distribution scheme described above will not only distribute and synchronize singular objects but it is possible to distribute parent child relationships between these objects too. Additionally, it replicates a complete scene graph to all processes in the same distribution group.
Figure 5.8 illustrates all steps of the process when distributing a scene graph using Avango. In a first step the first participant loads up an Avango scene represented by the scene graph including nodes, sensors and services. After a remote participant joins the distribution group using local sensors and services (step two), the scene graph is copied according to the hierarchical order of the nodes (step three). In step four the fields of the local scene graph nodes are copied into the corresponding fields of the remote scene graph nodes which turns the remote scene graph into an identical copy. Now the remote sensors are connected to the nodes which allows interaction at the remote side as well (step five). As soon as these interactions change the state of the own nodes, represented by their field values, these changes are propagated back to the first participant again (step six). Step six occurs every time one participant in the distribution group interacts in the Virtual Environment coursing changes of the scene graph state.
Being able to distribute the entire scene graph with all parent-child relationships between the objects is a key feature on the way towards a simplified development of distributed applications. However, it is not sufficient when considering the case of a distribution group with two member processes A and B. Both processes have already created several distributed objects in that group. Now a third process C joins the group. From now on all three processes will be notified of future object creations and manipulations, but process C will not know of the objects that A and B already created before it joined.
Avango overcomes this problem by performing an *atomic state transfer* to every joining member. When a new process joins an already populated distribution group, one of the older group members takes the responsibility to

transfer the current state of the distribution group to the new member. This involves sending all objects, currently distributed in the group, with all their field values to the newcomer. After the state transfer the new member will have the proper set of object copies for this distribution group. To prevent consistency problems, the state transfer is performed as one atomic action by suspending all other communication while performing it.

The ability to replicate the entire scene graph paired with the state transfer to joining members effectively solves the duplicate database problem. New members can join an existing distribution group at any time and will immediately receive their local copy of the scene graph constructed so far in the distribution group. Furthermore, the application programmer does not need to be concerned with distribution details. The user can take the scene graph for granted on a per process level and can concentrate on the semantics of the distributed application. Figure 5.9 shows a brief scheme code example distributing a simple jaw bone geometry.

```
;; join the distribution group with the help of
;; string "codeword"
(define dist-group-node (av-join "codeword"))

;; instantiate a file node to load the geometry
(define geom (make-instance-by-name "fpFile"))

;; load the geometry (-> (-> geom 'Filename) 'set-value
         "http://viswiz.gmd.de/~gernot/graphics/iv/jaw_bone.iv")

;; distribute the geometry
(-> dist-group-node 'distribute-object geom)

;; after distribution add geom to the distribution group node
(-> (-> dist-group-node 'Children) 'add-1value geom)
```

Figure 5.9: The call of the `av-join` command creates a distributed session. This session can be joined by using the *"codeword"*. The instance of `fpFile` and thus the geometry can now be distributed applying *'distribute-object*. Joining Avango applications can access the geometry over the *URL* specified in the *'Filename* field since all fields are copied after the whole scene graph has been replicated. *geom* is attached as child to the distribution group node after being distributed.

5.2.2 Audio/Video Conferencing

The audio/video conferencing runs independently from Avango and thus independently from the scene graph distribution. In addition the video software sends and receives its streams independently from the audio source. However, the audio as well as the video setup show almost the same configuration. The flow chart in Figure 5.10 shows the single procedure steps of server and client. On the server site the frames are grabbed from the camera which is plugged

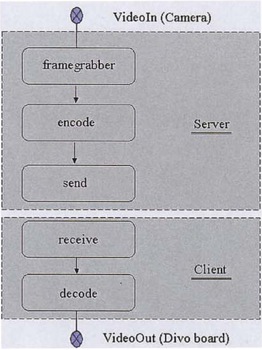


Figure 5.10: The figure shows the work flow of the video server and client. On the server side the frames are grabbed, encoded and then sent to the client side where the stream becomes decoded. The same sequence is used for the audio connection that runs independently from the video.

into the *VideoIn* port of the video card. Then the grabbed frames are encoded. After encoding, the reduced frames are packed into buffers and sent over the network to the client site that is already waiting. There the buffers are received and decoded. The inflated frames are then available at the *VideoOut* port of the O2 video card again.
The video setup decides about the next procedure. As shown in Figure 5.3 the video stream can either be chroma keyed or not.
Finally the video stream is transferred to the DIVO video board of the ONYX workstation. There the video is integrated and rendered in the CVE making

use of the Avango scene graph. Therefore the video frames are handled like a texture which can be mapped to a polygon representing the virtual video screen. Special video configurations in Avango download video frames from the video hardware continuously into the texture memory.
For sending and receiving the video and audio streams SGI O2 workstations are used in a way as shown in Figures 5.1 and 5.2. The decision for using these type of machines was mainly influenced by the availability of the O2 MVP video cards. These special video cards are equipped with a special motion jpeg encoding chip. The cards in general as well as the compressor chip in particular can be configured using SGI's *Digital Media Development Environment (DMdev)* library [86].
The audio/video conferencing software is developed on the basis of this DMdev library. This software handles the following different parts, namely: network communication, video path, video node, image compressor, image parameters, encoding and sending of the images. As the receiving unit works equivalent to the sending unit it is sufficient to focus on the latter. The program flow chart in Figure 5.11 shows the different initialization steps. The first step is to configure the network and the communication. Therefore three parameters are important: the protocol, the port and the host name of the receiving unit. Then the video path is configured which creates a link with the connected camera. For doing so the API functions of the VL (SGI's Video Library) are used. The VL allows also to configure video paths to more than only one connected camera. The video path consists of two nodes, the source and the target node. The source node is the camera and the target node is the memory segment to store the image. Later in the process the frames are grabbed from this memory for encoding.
The attributes of the video nodes that need to be set are image format, image size (PAL), zoom factor, color space etc..
The selection of the image encoder and its configuration needs as to be done according to the encoding requirements. The following encoders are available:

- Apple QuickTime Animation - 'rle'
- Cinepark - 'cvid'
- Intel Video- 'IV32'
- H.261 - 'h261'
- JPEG - 'jpeg'
- MPEG1-Video - 'mpeg'

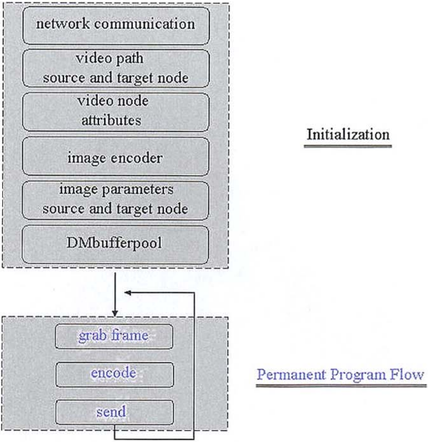


Figure 5.11: Program flow chart for the sending unit. The first block represents the initialization of the video board. The second block shows the permanent flow consisting of frame grabbing, encoding and sending.

The hardware encoder can use the jpeg standard. According to the requirements for the video conferencing the motion jpeg encoder is chosen. The parameters for this encoder type are set. The source and the target node are the uncompressed and compressed image respectively.
In the last step DMbufferpools are created. These pools are used to transport images on the video board. These pools are allocated in system memory to which all IO devices have a very fast access. Two DMbufferpools are created. One for the transport of the images from camera to memory (vid-to-mem) and one for transporting it to the image encoder. This last buffer is used again for writing the encoded image back into the same memory segment (mem-to-mem). With this the initialization phase is complete.
Then the permanent program loop is entered. This loop consists of the following three steps, grab a frame, encode and send. More precisely, every time a

new image is created at the memory target node an event is released. If this event (VLTransferComplete) occurs, the image frame is read into the DM-buffer and passed over (dmICSend()) to the encoder. dmICReceive() reads the compressed frame from the encoder again into the second DMbufferpool. Finally the call of dmNetSend() sends the compressed frame to the decoding client.
Since the video communication mainly occurs between the camera, the hardware encoder and the system's memory, the system CPU is mostly spared. Transfer and frame rate measurements showed a CPU load of less than 10% on a SGI O2 workstation with a MIPS R10000 processor. The separately handled audio connection adds another 5-7% to this CPU usage. These measurements are made having a video frame rate at the client side between 20-25 fps. The bandwidth necessary to ensure this transfer rate has to be at least 500 kbps. Similar results can be achieved using PCs with common video boards. VL- and DMdev-like digital media libraries are also available for PC hardware. The concept, however, remains the same.

Stereo-Video Conferencing

The stereo video conferencing is especially challenging because of two things:

- grabbing, encoding and transferring two synchronized images
- integrated rendering of synchronized stereo textures in Avango

When transferring two synchronized images, that correspond to each other, it has to be ensured that both images arrive completely at the other site. One image only is of no use. Due to this requirement the fast UDP protocol cannot be considered as it offers no confirmation mechanism. The TCP protocol instead is an acceptable option. An optimal solution for the stereo video conferencing is to send the images together at once and not after each other. Additionally this implies that both images, the one for the left and the one for the right eye, should be grabbed together as a mixed image from the camera. Here mixed means that both images share the 576 lines (fields) of the PAL sized images. How this could look like is shown in Figure 5.12.
It is possible to configure the video nodes in a way that only the odd fields are grabbed from the right camera image and the even fields of the left camera image. Then both images, having half size only, can be merged. This merging can either be done so that the upper part of the image is of the right camera and the lower part of the left camera or they both are merged alternating like shown in the right hand side of part of Figure 5.12. Merging half-images of both cameras has the advantage that the amount of data is the same as when using mono video conferencing. It is evident that extra CPU time needs to be

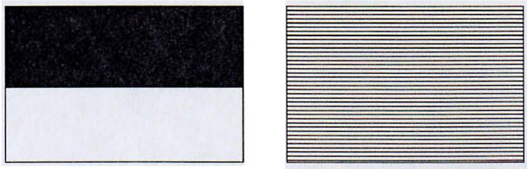


Figure 5.12: The video server can be configured to grab the even fields of the right and the odd fields of the left camera. Then both fields are assembled together in one image which will again be encoded and sent to the client. With this trick it is possible to keep a high frame rate even sending stereo images over the network.

spend to copy the half-images into the same DMbuffer on server and client site. Additionally it ensures a PAL sized mutual image which can be compressed by the video hardware which is optimized for PAL sized images only. As soon as the image resolution decreases, the encoding needs to be done by software which would result in a decreasing frame rate.

After the stereo images are received by the client site and are decoded and splitted again they have to be texture mapped and rendered in the CVE. This integration into Avango is quite challenging as the draw traverser does not know anything about the synchronized images which arrive at the DIVO video hardware of the ONYX workstation (see Figure 5.3). For solving this problem a mechanism is created which knows about the synchronized video images and decides which one of them is rendered when. As already said video images are handled like textures which are permanently downloaded from the video hardware into the ONYX texture memory. Thus the geometry onto which these textures are mapped has to be added and removed to and from the scene graph according to the framerate. For doing this a fpDrawEyes node is implemented which offers two special methods, a pre_draw_callback() and a post_draw_callback() (see Figure 5.13).

According to the frame count the pre_draw_callback() switches the geometry invisible through overriding the geometries material properties. This happens before the draw traverser renders the scene. After the rendering the post_draw_callback() switches the geometries visibility again through disabling the material override mode. When the draw traverser arrives again at the beginning of the next frame the pre_draw_callback() is not going to be executed as the Performer function pfGetFrameCount() increases the currentFrameCount by one only every second frame, as one frame consists of two images to be rendered (left and right eye).

Figure 5.14 shows the corresponding scene graph created by an Avango scheme script. The whole scheme script is shown in Appendix B.

Both fpLoadFile nodes (they are similar to the fpFile nodes) lay on top of

```
int fpDrawEyes::pre_draw_callback(pfTraverser* trav)
{
  long int currentFrameCount = pfGetFrameCount();

  if (( _whichEye && currentFrameCount == _oldFrameCount) ||
      (!_whichEye && currentFrameCount != _oldFrameCount) )
    {
     pfOverride(PFSTATE_TRANSPARENCY, PF_ON);
     pfOverride(PFSTATE_FRONTMTL,     PF_ON);
     pfOverride(PFSTATE_BACKMTL,      PF_ON);
    }
  _oldFrameCount = currentFrameCount;

   return PFTRAV_CONT;
}

int fpDrawEyes::post_draw_callback(pfTraverser* trav)
{
  pfOverride(PFSTATE_TRANSPARENCY, PF_OFF);
  pfOverride(PFSTATE_FRONTMTL,     PF_OFF);
  pfOverride(PFSTATE_BACKMTL,      PF_OFF);
  pfPopState();

  return PFTRAV_CONT;
}
```

Figure 5.13: Implementation of fpDrawEyes' pre_draw_callback() and post_draw_callback(). The pre_draw_callback() switches the geometry invisible before the rendering whereas the post_draw_callback() switches the geometry visible again after the rendering.

each other since they have the same matrix transform and share the same parent node. Due to the switching of the fpDrawEyes nodes only one of them is going to be visible at a time. In the real implementation they are a little bit tilted to each other according to the user's eye position. This ensures that each eye's view direction stands orthogonal to the screen with the texture of the corresponding camera.

5.3 Conclusions

This chapter described the CVE implementation details with respect to the used hardware and the software configuration. The hardware configuration

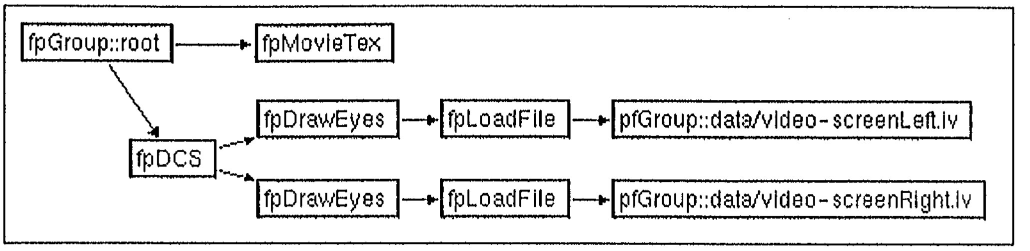


Figure 5.14: Scene graph showing the fpDrawEyes node. This node is responsible for switching between the two overlaying video textures.

is explained for a distributed setup using two Responsive Workbenches, and for a distributed setup using a Responsive Workbench and GMD's cave-like CyberStage. These two configurations include input devices for interaction, computers for rendering and distribution, and computers for video and audio streaming. Additionally it includes equipment like shutter glasses, infra-red emitters, cameras as well as microphones and headphones. This equipment can be used for any combination of display systems, including even more than two.

The software section described the Avango software framework which is used for rendering and distribution. Thereby Avango's field interface is introduced as well as the importance of field connections. Additionally it is explained how comfortable distribution mechanisms make use of this field interface in order to handle the database duplication problem. The remainder of this chapter describes the video conferencing software with respect to its mono and stereo video conferencing capabilities. A solution for sending synchronized stereo video frames is introduced as well as a solution for the integration and rendering of the synchronized stereo images in the CVE.

Chapter 6

Evaluation

Designers and programmers of Virtual Environments tend to guess about the best realization and implementation of interaction techniques or even whole applications. Many works have shown that user based assessment is an essential component of developing interactive applications and in this work it is shown that user based assessment is especially important for applications as complex and innovative as CVEs. Already the assessment of parts of the application by different users except the designers can substantiate or refute realizations of a specific CVE.

If those assessments are formalized they are called evaluations. There exist three different evaluation methods which can be applied to Collaborative Virtual Environments. The expert heuristic, the formative and the summative evaluation [53, 54, 74].

The *expert heuristic evaluation* is an analytical method. The evaluator is a field expert who determines problems with usability in the design phase of the CVE. It is important that the evaluating expert is not part of the developer group and not involved in the design of the CVE at all. In addition the expert has to assess the CVE as early as possible in the design phase and also has to determine problems during the entire CVE development cycle. Based on the expert's knowledge, problems concerning usability can be solved following the expert's recommendations. Especially when evaluating Collaborative Virtual Environments this process is of a particular challenging nature, because of the small number of VE experts worldwide. Additionally, there exist only a few VE user interface design guidelines and there is an absence of VE user interface standards.

The *formative evaluation* is an empirical, observational method. Thereby the evaluators assess the CVE throughout the entire development cycle. The output of this evaluation method is a combination of qualitative and quantitative results. The quantitative data evaluates the amount of time, the number of trials, the number of mistakes etc. while performing a special task. The qualitative data can be obtained by observing so-called *critical incidents* [53]. A critical incident is a problem that occurs while a user is interacting within the CVE. These incidents can be confusion, cancellation, errors, repetition etc. (3.2). Hereby the term critical incident does not necessarily mean that these events have to have negative effects on usability. Positive events can also have an impact on usability and thus user performance and satisfaction. These positive incidents contribute to qualitative evaluation results as well.

The *summative evaluation* is an empirical method. The objective of this evaluation method is to compare between different CVEs designed with the information obtained from the same User Task Analysis (UTA see section 3.2). Hence the output of the summative evaluation method enables the statistical comparison of different realizations of interaction techniques, operations, representation components etc. and the choice of the most appropriate one in terms of usability of the CVE. As this evaluation is performed using nearly final implementations of the CVE the evaluating users are usually those users the CVE has been designed for. However, a more important constraint is that the evaluators have to be non-experts in VEs and have not to be involved in the design process at all.

Best evaluation results can be obtained when combining the three methods described above. With respect to its nature the expert heuristic and the formative evaluation method should be applied in the early phases of the design process. Already short alternating cycles of these two methods can eliminate the biggest problems concerning usability and user satisfaction. For the assessment of more subtle differences in realizations and implementations of CVEs summative evaluation is absolutely essential. However, the most important and often most complex part to manage while planning an evaluation is to determine items to be assessed. This collection of items is necessary to formulate specific questionnaires and hence to find and eliminate disturbance factors within the implementation of the CVE.

6.1 Evaluation of H-C-H Interaction

In order to determine the evaluation items mentioned above the Human-Computer-Human model introduced in chapter 3 is very helpful. As a reminder, in that chapter the three flows within the model have been determined. These are the $H \rightarrow C$, the $C \rightarrow H$ and the $H \leftrightarrow H$ via C flow (see Figure 3.1). The design objective is to enable Human-to-Human interaction in a CVE as if face-to-face where the computer as the mediator for this collaboration becomes omnipresent and transparent. From this point of view it is clear that each user has to perceive itself and the remote partner being as present as possible in the Virtual Environment. Although the perception of presence is not the only requirement for a good collaboration at least it is the basic constraint for the establishment of the latter. But also appropriate representation of information is important for supporting collaboration. Hence when talking about possible problems and bottlenecks of the Human-Human interaction, the $H \rightarrow C$ and the $C \rightarrow H$ flows are the originators of those problems. For example the $H \rightarrow C$ flow can be disturbed through one or even more of the following disturbance factors:

1. unsuitable graphical and physical user interfaces
2. unsuitable physical input devices and equipment for generating user input
3. unsuitable representations of actions and events

Factors that might disturb the $C \rightarrow H$ flow include:

1. slow data processing and system reaction time
2. low network transfer rate and network drop outs
3. low graphical and acoustical quality

Hence, collaboration can be supported by the designers of the CVEs only if weaknesses such as the ones listed above are eliminated. Back to the assessment of the CVE these considerations imply that the disturbance factors have to be evaluated in order to find the best realization with respect to the User Task Analysis (UTA). However, the disturbance factors listed above are very generic and not linked to CVEs. These factors can be used as macroscopic parameters only. To arrive at a more pragmatic formulation of the disturbance factors, the items above are matched with the taxonomy from section 3.1 and the Awareness-Action-Feedback loops from section 3.6. The resulting evaluation items for the $H \rightarrow C$ flow are respectively:

- unsuitable menu representations (the user does not know where to find the desired function)
- unsuitable tool representations (the user does not know which tool has which functionality)
- unsuitable representation of data and its functionality (the user does not know how to process things and how to fulfill the task)

- unsuitable environmental representations (the user is confused by the surroundings and cannot concentrate on the task or transfer the learned skills to real-world applications)

- unsuitable input devices (the user is unable to work with/handle the input devices and to generate input with them)

- unsuitable physical equipment and annoying cabling (the user is confused by the cabling of input devices and shutter glasses and thus is not acting naturally or is unable to concentrate on the task)

For the $C \rightarrow H$ flow the comparison leads to the following evaluation items:

- real-time system reaction (the user selects items from menus or performs changes on the data set and sees intermediate reaction of the system without perceptible time delay)

- low graphical and acoustical resolution and quality (the user is unable to recognize tools, data set structures, actions of the remote partner and the partner itself, click and warning sounds, or is unable to talk to the remote partner)

- low network transfer rate (the user is unable to recognize the delayed actions of the remote partner, there are interruptions of the audio communication, the user is unable the map actions of the remote partner with occurrences of action feedback such as highlights, click sounds, movements of data sets)

The items mentioned above have great impact on collaboration and can be extremely disturbing to the Human-Human interaction. However, in addition to these items there exist other factors that might also have an impact on collaboration. The character of these factors is based on personal perception of collaboration. In this work evaluation items linked to personal perception are defined that have a quantitative or qualitative nature like :

- perception of the own presence within the CVE

- perception of the partner's co-presence within the CVE

- perception of the collaboration in terms of equality of rights

- perception of the quality of collaboration

- frequency with which the user looked to the partner

- frequency with which the user spoke with the partner

Considering all these evaluation items in one session is almost impossible. The reason is that these items evaluate too many different aspects of the Human-Computer-Human interaction. In order to address this great amount of items special evaluation sessions had to be defined which are able to let assess specific aspects of Human-Human collaboration as it has been defined in the *H-C-H* interaction model.

6.2 Evaluation Sessions

According to chapter 4 three different sessions are implemented that in this chapter are used for extensive evaluation of the interaction taxonomy model and to produce CVE design guidelines. These sessions are:

1. usability session
2. co-presence session
3. co-work session

Before the evaluation an initial session introduction to the system is included. The introduction session itself is not part of an evaluation session. As the evaluators must not be the developers and must not be familiar with Virtual Environments they have to be introduced to VEs. During this introduction they are informed of the display system, the equipment and the environment they are going to work with. The objective and advantage is that this introduction session creates almost same conditions for all evaluators. This is necessary in order to compare numerical results of the summative evaluation. In order to exemplify the evaluation sessions the scenario described in chapter 4 is used in the following subsection.

6.2.1 Usability Session

The usability session is the first evaluation. After the users (evaluators) are introduced to the Virtual Environment they interact autonomously within the VE for about five minutes. During this interaction an observer is taking notes. Beside the overall ability to interact with the system the critical incidents of the formative evaluation are the most interesting to the observer. The application of the usability session offers almost the same interaction techniques, operations, tool representations, menus and feedback components to the user as in the following two sessions. Only the data set is exchanged in order to ensure that content specific operations have not been learned in this session already. If the observer gets the impression that the user is not yet familiar with the VE, the interaction time is extended. After the usability session is completed a questionnaire is handed out addressing usability assessments. After the questionnaire is completed the user gets another five minutes recovery time before starting the co-presence evaluation session.

6.2.2 Co-presence Session

The idea of this co-presence session is to evaluate the design of the CVE in terms of its support to the evaluator during a certain task using immersive telepresence only. As already mentioned earlier the perception of presence is not the only requirement for good collaboration but it is the basic constraint for the establishment of the latter. It is shown in the following sections that the evaluation results of the co-presence session are of high interest for distance learning applications.

In the co-presence session the user works again in the Virtual Environment but now with another data set. An experienced user who has been involved in the development process is present within the same environment using an remote audio/video connection.

The experienced user explains the task, the data set, the input devices and the tools remotely to the evaluator. Hereby the remote partner who acts like a supervisor does not use any input devices or tools. Only gestures and verbal instructions are used. The task is to position three bones as precise as possible in their correct location on a human female skeleton, as explained in chapter 4. These bones lie in front of the evaluator and look very similar to each other so that it is not obvious where they have to be add to the skeleton. If the evaluator does not know how to achieve the goal the supervisor gives advice about which tool should be used, how to query information about the bones, how to change the viewpoint etc. (see chapter 4). After the co-presence session is completed a questionnaire is handed out addressing co-presence assessments (see section 6.3.3). After the questionnaire is filled out the user gets another five minutes recovery time before starting the co-work evaluation session.

6.2.3 Co-work Session

The idea of the co-work session is to evaluate the design of the CVE in terms of its support for collaborative work and minimum time required to fulfill the task. Now it is important that both partners have equal rights concerning decisions, manipulations and the access to tools. It is interesting to note that in the implemented session both users can complete the task autonomously

as well. So the main question is to evaluate whether the CVE is capable of supporting and encouraging for team work although team work is not really required for this task.
The evaluator works in the VE and the remote partner is present using an audio/video connection again. Although the partner is one of the developers and thus an VE expert both are equal in rights and are using the same tools for interaction. The task is slightly different to the one of the co-presence session. Together the users have to position six bones belonging to three different pairs to complement the human female skeleton. Each bone in a pair belongs to the left or right side of the skeleton (i.e. the femur bone of the right and the left leg). A set of three of these bones lie in front of each user. As the users stand opposite each other on different sides of the skeleton they have to find out which bones belong to their side as the bones are mixed. This can be done by querying the name of the bone or by comparing them directly which is however more complicated. If a user finds out that a bone belongs to the partner this bone can be exchanged by passing it over to the other side. After a bone has been positioned in the skeleton the user can make use of a snap-back tool which lets the touched bone snap into the correct position. Thus it is possible to verify their own or their partner's work. In order to complicate the task the human female skeleton is covered by its own skin. For positioning the bones the particular part of skeleton has to be made visible by cutting away the skin in this region. For doing so one user selects a special cutting tool from the ring menu and apply this to the interesting part of the female's body. It is not possible to cut the skin in this region permanently. This means that the cutting user has to hold the skin cutter while the other user positions the bone. Instead of using a snap back tool to verify the position the user can grow the skin back and verify whether bones stick outside the body or not. After the six bones have been set into the skeleton correctly the co-work questionnaire is handed out (see section 6.3.4).

6.3 Evaluation Questionnaires

The different evaluation sessions are implemented because the evaluation items from section 6.1 assess too many different aspects of the *H-C-H* interaction model. Therefore questionnaires are developed to let the evaluators assess these different aspects.
The items of the questionnaires are enumerated. The usability questionnaire starts with *A*, the co-presence, co-work and observer with *B*, *C* and *D* respectively. All answers are ranged in the interval of 0 - 6. This is done in accordance with other evaluations [91, 92, 97, 103]. In order to support

the evaluator assessing the different aspects of interaction, descriptive text is placed beside the answer possibilities (i.e. 0 corresponds to never/bad/no, 3 to sometimes/acceptable/maybe and 6 corresponds to often/good/yes). The text based support makes it possible to place assessments on a numeric scala more precisely whereas the numeric results are necessary for the statistical analysis.

6.3.1 Introduction Questionnaire

Before the first evaluation session information about the user's profile is queried (appendix C). Interesting in terms of the evaluation is information about the evaluator's profession and experience with computers. Additional information on whether the evaluator is right or left-hander is asked. Early, external observations showed that it is not necessary to make a distinction between different genders. This work shows later again that only age and experience with computers rather than gender has an impact on the ability to get used to interaction in Virtual Environments in contrast to other researchers [92].

6.3.2 Usability Questionnaire

The usability questionnaire is handed out after the first evaluation session (appendix D). This questionnaire addresses evaluation items concerning the *H-C* interaction from section 6.1. Questions are listed assessing the quality and comfort of input devices, tool and device representations, positions of menus and the appearance of text in the VE.

6.3.3 Co-presence Questionnaire

The co-presence questionnaire queries information about evaluation items which have an impact on the perception of co-presence and on the communication between the two partners (appendix E). Social aspects of direct Human-Human communication are addressed here such as the influence of the size and shape of stereo glasses on communication and thus the exchange of information, as well as the position and the size of the partner's representation. In addition to that technical aspects are addressed such as the impact of network delays and drop outs on communication and collaboration.

6.3.4 Co-work Questionnaire

The co-work questionnaire is filled out after the last evaluation session is completed (appendix F). In this questionnaire information concerning the direct team work is queried such as the frequency with which the evaluator was looking at the partner. Additionally evaluation items concerning the perception of co-knowledge and co-status are of high interest (see section 3.6). Social aspects such as the perception of being equal in rights during the collaboration are queried too. Important is that similar questions are asked in the co-presence and co-work questionnaire. The driving idea behind this is the evaluation of differences related to the social aspects of collaboration. The character of these two questionnaires enables investigations on differences in perception of co-presence. This difference might exist between situations where partners communicate only in comparison to situations where they communicate and work together (see analysis section 6.4).

6.3.5 Observer Questionnaire

This questionnaire is different from the others as it is not corresponding to a particular evaluation session (appendix G). It is filled out by an external observer who is an expert. This VE expert is observing the non-expert evaluator during the usability, the co-presence and the co-work session. Besides querying specific information about the time the user had to think and to debate before performing actions the questionnaire leaves also space for informal observations. These informal observations correspond to the critical incidents of the formative evaluation. These incidents observed from outside give feedback about the abilities of the evaluators. They have great impact on the following statistical analysis of the numeric evaluation results. Especially this questionnaire helps assessing items which are difficult to be assessed by the evaluator itself such as questions D5 or D6:

- Did the user loose concentration during a session ?
- How quickly could the user correct mistakes and continue the work ?

Information whether the evaluator lost concentration during a session has an impact on the analysis and the way the numerical results have to be interpreted. However, this information can also imply the high cognitive load of interaction in the Collaborative Virtual Environment too.

6.4 Evaluation Analysis

The evaluation analysis focusses on the three different evaluation methods introduced earlier in this chapter. These methods are the expert heuristic, the formative and the summative evaluation. In the early CVE design phase alternating cycles of expert heuristic and formative evaluation are performed in

order to eliminate obvious usability problems from the very beginning. For obtaining more subtle results concerning usability and team work summative evaluation is applied.
The CVE evaluated by 60 people has passed through expert heuristic and formative evaluation already. It is improved before given to summative evaluation. Now, the focus of this section is on the analysis of summative evaluation results. The first evaluation results and the design guidelines are given in section 6.4.2.
For analysing the numerical data obtained by the summative method expectancy values $\overline{x_j}$ are computed. In order to handle the uncertainty of the numeric results the standard deviation s is computed from

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

where x_i is the value of measurement i, $\overline{x}$ the expectancy value and n the total number of measurements. s^2 is often also denoted as *variance*. Thus follows that the statistical certainty of the average value $\Delta\overline{x}$ is represented by

$$\Delta\overline{x} = \frac{t}{\sqrt{n}}s = \frac{t}{\sqrt{n}}\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

where the factor t is depended from the statistical certainty P and the total number of measurements n. Internationally in industry the statistical certainty is considered to be $P = 95\%$. Hence values for $\frac{t}{\sqrt{n}}$ are [65]:

$P = 95\%$	n	3	4	5	6	8	10	20	30
	t	4.3	3.2	2.8	2.6	2.4	2.3	2.1	2.05
	$\frac{t}{\sqrt{n}}$	2.5	1.6	1.24	1.05	0.84	0.72	0.47	0.37

However, before computing the expectancy values and their statistical certainty from the numerical results of the usability, co-presence and co-work questionnaires an analysis of the evaluators' profile is given.

6.4.1 User Profile

For the assessment of the CVE in expert heuristic and formative evaluation cycles two VE experts have been consulted who are not part of the CVE design team. Throughout the entire summative evaluation process almost 60 non-expert users have been evaluating the CVE. They worked together within the three sessions introduced in section 6.2.

Analysing the introduction questionnaire from appendix C a user profile is created. The age of the 60 evaluators is minimum 17 years and maximum 58 years. The majority are between 22 and 27 years old. Most of these evaluators are university students whereas the diversity of the others' professions reaches from secretaries and journalists over workers, technicians to technical and non-technical university professors and researchers. Although all evaluators are no Virtual Environment experts the knowledge concerning computer hardware and software differs substantially. The group of 22-27 years old uses the computer mostly for web surfing as well as computer games whereas the older evaluators use it for editing with text processing software. This is the reason why this first group is more experienced with hardware devices, such as game joysticks and steering wheels including force feedback. This observation is independent from the subject's profession or field of studies. A contrary result is that the older evaluators use a computer almost twice as long per week as the group of the 22-27 years old. No other significant differences between the evaluators that might have an impact on the analysis of the evaluation results are found.

6.4.2 First Level Analysis

The first level analysis is split into two parts. The first part deals with the results obtained by the expert heuristic and formative evaluation. Although the outcome is taken into account already it is quite instructive to discuss it separately from the results of the summative evaluation.
The usability findings and recommendations of the expert evaluators concern the following items [50]:

1. positioning of the toolbar grouping generic operations
2. handling of the ring menu grouping content specific operations
3. tool representations on menus
4. three button input device and stylus vs. pinch gloves
5. ego-centric vs. exo-centric viewpoint manipulation
6. graphical representation of data set
7. video frame rate

The User+Need Space (UNS) for the considered evaluation scenario of chapter 4 determines different representation forms for generic and content specific operations. For the generic operations a toolbar is developed whereas the content specific operations are grouped by a special ring menu. In early designs of the CVE the generic toolbar was configurable in position by the user. The idea behind was that a dominant right-handed user might want to position the menu somewhere else in space than a dominant left-hander. Evaluation results showed that configuration of menus has a negative impact on the cognitive load. Additionally it is not really used in limited interaction spaces offered for example by the Responsive Workbench (RWB). Working with both hands at a RWB, the total viewing frustum is accessible in contrast to CAVE-like display systems. Thus during the formative and summative evaluation the toolbar was positioned close to the users body within arm distance corresponding to the vendor's tray metaphor. Working at a RWB this toolbar is fixed whereas it is attached to the user's body position when working in a CAVE or cylindrical and wall display systems.

Similar problems are encountered when using ring menus described in [49]. When a user intersects the data with the menu pick ray in the right hand the ring menu appears attached to the left hand and vice versa. This corresponds to the metaphor of handling a painter's palette with respect to dominant right and left-handers. The advantages were assumed to be the comfortable handling of this ring menu since it does not occlude any object being handled this way. For detaching the ring menu, over the shoulder deletion was integrated (see Figure 2.9). Evaluation results showed that the handling making use of the painter's palette metaphor is not always as comfortable as assumed. The reason is that the user first has to recognize that the status of the hand changed as something is suddenly attached to it. Then the user has to look at the ring menu in order to select a content specific operation using the other hand. This is particularly annoying if the hand is busy with another task already. Additionally this metaphor makes it impossible to concentrate on the data set as the user is forced to turn the head towards the ring menu. In the improved design the ring menu is attached to the calling hand holding the menu pick ray. It follows the translation of the user's hand whereas the rotation of the user's wrist is used to intersect the ring pieces with the pick ray. The advantages are that the menu appears within the user's gaze and disappears as soon as the user releases the stylus button again. The menu is designed to be 70% transparent to avoid occlusion of data (Figure 3.6).

As already mentioned the menus group operations together. In order to apply operations, tools are selected, e.g. the zoom operation requires a special zoom tool. The tools are represented by 3D icons which are attached to the buttons of the toolbar or to the choices of the ring menu. Usability findings showed that representations for the snap back tool, the information tool and the skin cutting tool were not appropriate in the early CVE design. Now the snap back tool is represented by a three dimensional hook icon, the information by a three dimensional "i" letter icon and the skin cutter tool by a three dimensional knife icon. These virtual tool representations increased the evaluator's tool recognition rate by almost 80%. Evaluation results indicated also that early approaches using two pinch gloves as input devices were not really addressing the user's needs. Reasons are the uncomfortable usage when working stand-alone collaboratively and trying to hand over pinch gloves to another user. Another encountered problem using pinch gloves together with pick rays is that it is almost impossible to keep pointing somewhere and additionally snap with the middle finger and the thumb for selection. Similar problems using pinch gloves have been encountered in [54]. Improvements are made by using a special three button tool in one hand and a stylus in the other. The reason for not using three button tools in both hands refers to the high cognitive load of their usage due to the many buttons. After modification evaluation showed that the stylus is rather used in the dominant and the three button tool in the non-dominant hand.

A *sharing viewpoint* metaphor is implemented for manipulating the users' viewpoint [49] (see chapter 4). Evaluation results showed that an exo-centric viewpoint manipulation is better than an ego-centric when *standing almost beside the partner*. In this context exo-centric manipulation is based on how a user would act in real world by moving laterally. When sharing the same viewpoint (*looking through the partner's eyes*) or sharing the mirrored viewpoint (*looking from opposite the partner*) ego-centric viewpoint manipulation is implemented. This manipulation is realized by pressing and releasing a special button on the three button tool. These observations are valid working at a Responsive Workbench. Because of the limited interaction space it is possible to access the data set visually from all sides by manipulating the viewpoint as described above. However, other own evaluations showed that in the CVE implemented using a CAVE and a cylindrical display no ego-centric viewpoint manipulation is needed. Here users prefer exo-centric viewpoint manipulation due to the larger interaction space and the perception of entire immersion.

In the co-work session the evaluators complement a female skeleton by missing bones (section 6.2.3). There the task is aggravated as the skin of the body is cut in order to make the skeleton visible. Usability findings indicated that users prefer to get a quick overview of the situation. This leads to the implementation of a content specific wireframe operation. The users are able to only render the skin of the body in wireframe and thus have a direct view onto the underlying skeleton. With this, strategies can be discussed and collaborative tasks can be planned more quickly. This content specific wireframe operation is only usable for getting an overview. For complementing the skeleton the skin has still to be cut.

In addition, observations of critical incidents are made during the co-presence session. These critical incidents occur due to network drop outs, indicating that the perception of co-presence is interrelated with the video frame rate. Further experiments with the video frame rate as parameter showed that the perception of co-presence vanishes completely if the video frame rate sinks below 12 fps.

Statistical Analysis

The next phase of the first level analysis is the statistical analysis of the summative evaluation results. In literature the analysis of numerical data is often restricted to the direct comparison of calculated statistical values [92]. In order to correlate with the work of other investigators direct comparison is performed [103]. Computing the average (expectancy) value and its statistical certainty $\overline{x} \pm \Delta\overline{x}$ for the usability questionnaire from appendix D leads to the following values.

Question	*A1*	*A2*	*A3*	*A4*	*A5*	*A6*	*A7*	*A8*
$\overline{x}$	**5.4**	*4.7*	*4.2*	**3.8**	*4.6*	*4.6*	*4.9*	*4.7*
$\Delta\overline{x}$	**0.54**	*0.68*	*1.05*	**0.99**	*1.05*	*0.77*	*0.76*	*0.79*

A conspicuous feature of the average values is their low statistical certainty represented by the high values for $\Delta\overline{x}$. This is especially the case for low average values like they appear for questions A3, A4 and A5. Obviously one reason is that the statistical certainty of the average value is dependent from the number of evaluators. The larger the number of evaluators the more certain the expectancy value $\overline{x}$ and thus the lower $\Delta\overline{x}$. But there is another reason why the statistical certainty is small for high average values and vice versa. The distribution of answers to questions with small average values is more spread whereas the distribution of answers to questions with high average values is sharper. For example the distribution of answers to a question with an average value of "3" might look like: 2 times "0", 2 times "3", 2 times "6". Here the answers are spread and the statistical certainty is very low. An example for the distribution of answers to a question with an average value of "5" might look like: 2 times "4", 2 times "5", 2 times "6". In this example the answers are distributed much closer around the true expectancy value of "5" and thus the certainty of the average value is higher. In any case, the spread distribution of answers to a question indicates that the evaluators have different opinions about the particular evaluation item. Although it is much more expressive to have a uniform answer pattern the spread distribution of answers implies problems with usability. If so, more investigations concerning the special evaluation item have to be made.

However, the average values represent important information about usability too. The highest average value with corresponding high certainty is computed for question A1 rating the responsiveness of the CVE (see appendix D). High values are also found for A2 (usability of input devices), A7 (alignment of tools) and A8 (alignment of menus). Lowest average values but with low certainty are computed for A3 (working with stereo glasses) and A4 (working with the cabling). As discussed above the statistical values for A3 and A4 indicate that not all evaluators perceived the work with stereo glasses and cabling as especially uncomfortable. The analysis of these special usability assessments in conjunction with the analysis of the external observations delivered no further results. From this, preliminary conclusions are drawn, indicating that the bad assessment of the stereo glasses and cabling is mainly influenced by personal perception rather than general frustration with the system or even inability to use the CVE.

Proceeding with the analysis of the co-presence questionnaire the following statistical values are computed:

Question	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*	*B8*	*B9*	*B10*	*B11*
$\overline{x}$	*5.1*	*4.6*	*4.2*	**5.5**	*5.1*	*4.9*	*4.6*	*3.8*	*3.8*	*4.3*	**3.1**
$\Delta\overline{x}$	*0.83*	*0.77*	*1.22*	**0.52**	*0.77*	*0.88*	*1.05*	*0.92*	*0.71*	*1.07*	**1.35**

The average values representing the answers to questions B1 to B7 are notably high. For example most of the evaluators perceived the partner standing as a real person (B4) on the other side of the table (B1). Also the partner's stereo glasses did not have a very high impact on the perception of co-presence (B2). Remarkably are the average values for questions B5 and B6. It seems that evaluators did not perceive the remote partner to be less present when delays in video and audio transmission occur. Additionally B7 and B8 show that evaluators think the audio and video representation of the partner are necessary to complete the task even though the partner was not looking very often to some of the evaluators. This is indicated by the high uncertainty of the B2 average value. In comparison to this it seems contradictory that B10 shows an average value of about $\overline{x} = 4.3 \pm 1.07$. B10 is assessing a higher transfer rate to be an important factor for increasing co-presence. Very interesting is the high uncertainty $\Delta\overline{x}$ of question B11. B11 is assessing the position and the size of the partner to be an important factor for increasing co-presence. Observations showed that especially evaluators who were much taller or smaller than the remote partner in reality thought that this parameter increases the co-presence. This explains the spread distribution of answers to B11.

Computing the statistical values for the questions of the co-work questionnaire results in the following table:

Question	*C1*	*C2*	*C3*	*C4*	*C5*	*C6*
$\overline{x}$	*5.7*	**5.8**	*5.0*	*5.2*	**3.1**	*5.1*
$\Delta\overline{x}$	*0.49*	**0.46**	*1.02*	*1.06*	**0.86**	*0.53*
Question	*C7*	*C8*	*C9*	*C10*	*C11*	*C12*
$\overline{x}$	*5.6*	*5.3*	*4.0*	*4.7*	*4.1*	*5.5*
$\Delta\overline{x}$	*0.70*	*0.76*	*0.83*	*1.18*	*1.10*	*0.61*

Average values assessing collaboration are remarkably high in comparison to the sessions A and B. The evaluators assessed the co-work session as a satisfying (C12) event where they really collaborated (C7) with a real person (C1) of equal rights (C8). This assessment is a very interesting result as it seems to disagree with the low value of question C5. Here the evaluators confirm that they were not looking often to the remote partner although they pretended to know always where the partner stood (C4). Which seems a contradiction at first, but becomes explicable when taking the assessment of question C9 into account. Here the evaluators state with an average of $\overline{x} = 4.0 \pm 0.83$ that body and hand gestures did not greatly enhance or support the collaboration. This leads to the preliminary conclusion that the actions and behaviour of the remote partner are adequately rendered by the representations of tools and input devices together with acoustic feedback. This is also underlined by the average values of questions C10 and C11 where the evaluators think they could have completed the task perhaps even without the video representation of the partner.

6.4.3 Simple Guidelines

With the help of the evaluations done so far it is possible to draw and summarize some conclusions.

Highlights of Expert Heuristic and Formative Evaluation

From the analysis of the expert heuristic and formative evaluation cycles the following guidelines can be extracted:

- Static menus should be simple, well positioned and non configurable. They are well suited for grouping generic operations.

- Dynamic menus should appear within the user's gaze (view) and must not occlude data. They must be designed in a way that the user is able to concentrate on the task while using the menu.
- During the task intuitive, recognizable tool representations are necessary. Additionally they are able to render the actions of the remote partner adequately.
- Input devices chosen should be easily passable from one user to another. Input devices should preferably support both, left and right handed users. Additionally they must allow precise application of an operation.
- Possibilities to get quick information or even an overview of the current situation in the CVE must be provided (wireframe technique, World in Miniature technique (WIM).
- Depending on the amount of interaction space and consequently depending on the display system intelligent partitioning of ego-centric and exo-centric viewpoint manipulation is necessary.
- Using immersive telepresence the video frame rate must not sink below 12fps.

Highlights of Statistical Analysis

From the statistical analysis of the evaluation results it is quite difficult to extract guidelines as they reflect the assessment of realizations by the evaluators. However, with the analysis of conspicuous distributions of answers to special questions it is possible to obtain guiding information.

- Cabling of input devices, trackers and stereo glasses are perceived as annoying. Careful handling of loose wires is recommendable.
- In a consultation situation immersive telepresence supports the work flow. In this situation network drop outs do not have negative impact on the perception of co-presence as long as the average frame rate does not go below 12fps.
- In a collaboration situation using immersive telepresence the position of the remote partner representation should be chosen in a way that both partners seem to have same virtual size in the CVE independent from their physical size in real world.
- Appropriate representations of the remote user's tools and input devices support collaboration more than body and hand gestures.

- When using a RWB the perception of co-presence can be increased with a remote partner's video texture representation together with a real background since due to depth perception the user has the impression that the remote partner stands closer to the table.

- When using a CAVE-like display system or cylindrical projections a remote partner's video texture representation without background is recommendable. This is possible, using a uniform background behind the user and a hardware keyer which subtracts this uniform texture pattern from the users outline. Using a real background together with the video texture of the remote partner leads to a mismatch of the non-photo realistic virtual scenery and the real video background (see Figure 5.3).

6.4.4 Group Analysis

Direct comparison of average values and their statistical certainty is a good method for getting initial quick results. In order to obtain more detailed information about usability of the CVE this method is not sufficient. Additionally, for investigating the problem of spread answer distributions another method than the direct comparison needs to be considered.

Human communication serves as an exchange of information. Independent of the appearance of this information the principle basis for communication is alternating cycles of query and answer. On the one hand queries are conceivable which only have one intention or at least expect one precise answer. On the other hand there exist more subtle queries which have more than one intention. According to evaluation it is possible to extract more than one intention from the items of the questionnaires. Within this thesis questionnaires are designed in a way that the same evaluation item could be evaluated by more than one question of different evaluation sessions. With the help of these special questions the following dependencies of evaluation items for the $H \rightarrow C$ and $C \rightarrow H$ flow are decoded:

- Perception of the system's responsiveness and its impact on collaboration (A1, B1, C1, C3)

- Impact of stereo glasses, cabling and input devices on collaboration (A2, A3, A4, B2, B3, D3, D4)

- Perception of tools and tool representations (A6, A7, C3, D1)

- Perception of menus and functionality
  (A5, A8, D1, D5, D6)
- Impact of audio/video transfer rate on co-presence and collaboration
  (B1, B5, B6, B10, C5)
- Perception of telepresence and its impact on collaboration
  (B6, B7, B8, B9, B10, B11, C5, C6, C9, C10, C11, D4, D6)
- Perception of co-knowledge and collaboration
  (B1, B4, C1, C2, C5, C6, C7, C8, C12, D2, D7)

With these decoded dependencies it is now possible to analyse the different evaluation sessions again.

Statistical Analysis

Comparing the average values and their statistical certainty of the questions A1, B1, C1 and C3 leads to the following table.

Question	*A1*	*B1*	*C1*	*C3*
$\overline{x}$	*5.4*	*5.1*	*5.7*	*5.0*
$\Delta\overline{x}$	*0.54*	*0.83*	*0.49*	*1.02*

All four questions assess the perception of system responsiveness, either explicitly or implicitly. In this group comparison question A1 directly queries information about the responsiveness of the system. The questions of the co-presence session (B) and the co-work session (C) assess the system's responsiveness implicitly. These questions query information about the perception of collaboration and co-presence explicitly. Remarkable is that all the average values and their statistical certainty highlight the system's responsiveness equally good and its impact on co-presence and collaboration as positive.

Questions A2, A3, A4, B2 and B3 evaluate the impact of hardware components such as input devices, cabling and stereo glasses on collaboration.

Question	*A2*	*A3*	*A4*	*B2*	*B3*	*D3*	*D4*
$\overline{x}$	*4.7*	*4.2*	*3.8*	*4.6*	*4.2*	*5.4*	*5.2*
$\Delta\overline{x}$	*0.68*	*1.05*	*0.99*	*0.77*	*1.22*	*0.69*	*0.76*

The usability ratings of the evaluators are average. The influence of hardware equipment on collaboration is assessed as average. The diversity of answers is expressed by the high uncertainty $\Delta\overline{x}$ of A3 (*comfort of the stereo glasses*) and B3 (*was your partner looking at you*). In accordance to these assessments the observer questions D3 and D4 confirm that the evaluators had

problems selecting tools and problems with orientation. Here the statistical certainty of observer evaluations are average.
Similar evaluations are encountered for the representations shown in the first part of the following table:

Question	A6	A7	C3	D1
$\overline{x}$	4.6	4.9	5.0	3.2
$\Delta\overline{x}$	0.77	0.76	1.02	0.73

Question	A5	A8	D1	D5	D6
$\overline{x}$	4.6	4.7	3.2	4.6	4.8
$\Delta\overline{x}$	1.05	0.79	0.73	0.59	0.85

The evaluators assessed the tool representations bit better than average suitable (A6) and their alignment as more or less appropriate (A7). With C3 the evaluators confirmed to know about what the partner was doing in the collaboration session which is ascribable to the tool representations. However, the high uncertainty of C3 indicates that some evaluators knew about the partner's actions but others did not. The time which was required to think about an operation before applying is assessed by the observers to be average with a normal certainty value.
The second part of the result table shows the ratings of A5, A8, D1, D5 and D6. Users evaluated the appearance of text (A5) and the alignment of menus (A8) to be a bit better than average but not as good. Informal observations showed that the high uncertainty value for the assessment of text in the CVE is mainly influenced by the physical size of the users. Users which were smaller than the average user had problems to read the text due to the declination of their viewpoint. Obviously text is not readable from any point in the interaction space. The assessment from the observers imply that the evaluators did not have to think too much about performing a certain action implied by problems with the menus (D1). But remarkable is that observers notice a loss of concentration during the sessions (D5). Even more, the highest assessment in this context is encountered for question D6. Here the observers confirm that the evaluators were able quite quickly to correct mistakes and continue the work.

The following table shows statistical values for questions rating the impact of audio/video transfer rate on co-presence and collaboration.

Question	B1	B5	B6	B10	C5
$\overline{x}$	5.1	5.1	4.9	4.3	3.1
$\Delta\overline{x}$	0.83	0.77	0.88	1.07	0.86

Evaluators confirmed that they perceived the remote user as standing on the other side of the table. Even a delay in video and audio transfer rate did not have a negative impact on immersive telepresence and thus on the perception of co-presence (B5, B6, B10). As described in chapter 5 the audio and the video streams are not synchronized. This does not seem to have negative impact either. Experiments with the network delay showed that this is valid as long as the transfer rate is high enough to guarantee smooth movements. The magic threshold is 12fps. This result is remarkable as most of the evaluators confirm that they did not look often to the partner during collaboration. From this the conclusion is drawn that the video representation of the remote partner serves rather psychological than direct support purposes. This point is discussed in section 6.4.6 again.

The following tables show even more decoded dependencies than the ones above.

Question	*B7*	*B8*	*B9*	*B10*	*B11*	*C5*
$\overline{x}$	*4.6*	*3.8*	*3.8*	*4.3*	*3.1*	*3.1*
$\Delta\overline{x}$	*1.05*	*0.92*	*0.71*	*1.07*	*1.35*	*0.86*
Question	*C6*	*C9*	*C10*	*C11*	*D4*	*D6*
$\overline{x}$	*5.1*	*4.0*	*4.7*	*4.1*	*5.2*	*4.8*
$\Delta\overline{x}$	*0.53*	*0.83*	*1.18*	*1.10*	*0.76*	*0.85*

The perception of telepresence and its impact on collaboration is assessed implicitly by the questions B7, B8, B9, B10, B11, C5, C6, C9, C10, C11, D4 and D6. The table shows a very uniform answer pattern. Most of the average values are distributed into the interval between 3.0 and 5.0. The evaluators have the opinion that the video representation of the remote partner is necessary with an average value of $\overline{x} = 4.6 \pm 1.05$ (B7). Here the statistical uncertainty is quite high. On the other hand the evaluators said that maybe they are able to complete the task even without a video representation of the remote partner (B8). Again this result is accompanied with a high statistical uncertainty.

In accordance to the evaluation results, the perception of co-presence cannot really be improved by better lighting or positioning/scaling the representation of the partner (B9, B11). The audio/video transfer rate is perceived as a factor that might improve the situation (B10). It is conspicuous that the evaluators spoke more often (C6) than they looked at the partner (C5) during the collaboration session. Here the questions C10 (necessity of the partner's video representation) and C11 (ability to complete task without partner's video representation) are asked again (same with questions B7 and B8). The ratings are nearly the same as in the co-presence session which improves the certainty of the result as their statistical certainty of the average values is quite high with 0.92 to 1.18. Additionally C9 assesses the benefit of collaboration from hand and body gestures to be lower than expected. The observers confirm with D4 and D6 that the evaluators were able to correct mistakes and return quickly to work. The users did also not have significant problems with orientation and therefore the following conclusion is drawn.

The video representation of the remote partner has a psychological impact on the perception of co-presence, co-knowledge and collaboration. This means that the partner is contacted visually, (looking at him/her) only in the case of problems. Most of the time the audio connection and appropriate mapping of tool and input device representations seems to be much more important than the video connection. Informal observations emphasize this reflection. Section 6.4.6 deals with this problem again.

The group analysis of the perception of co-knowledge and collaboration is interesting as well as challenging. The following table shows the statistical results:

Question	*B1*	*B4*	*C1*	*C2*	*C3*	*C4*	*C7*	*C8*	*C12*	*D2*	*D7*
$\overline{x}$	5.1	5.5	5.7	5.8	4.6	5.2	5.6	5.3	5.5	3.1	6
$\Delta\overline{x}$	0.83	0.52	0.49	0.46	0.77	1.06	0.70	0.76	0.61	0.72	6

The high average values indicate that the collaboration is perceived as quite convincing. The remote partner was perceived as personally (B4,C1) standing at the opposite side of the table (B1). The collaboration was satisfying (C12) since most of the evaluators would like to work with the same remote partner again (C2). During the collaboration they perceived themselves as equal partners (C7) and also treated the partner as a person equal in rights (C8). Although the values are a bit lower than the average the evaluators knew where the partner was standing (C4) and what the partner was doing (C3) during the collaboration session. This values are confirmed by observer assessments of the amount of time the partners had to debate before they came to an agreement about a particular action (D2). The average value of 3.1 indicates that the time needed to do this is average. A clear "Yes" rating is given for the fun the users had while working in the CVE.

6.4.5 Advanced Guidelines

With the help of the group comparisons it is possible to draw some very important conclusions. From this the following guidelines can be extracted:

- Ensure high system responsiveness. It is perceived as having very positive impact on collaboration. Even downsizing the application in order to decrease the CPU load is recommendable. A good system responsiveness is guaranteed if all inputs and outputs are processed and rendered within less than 50ms.
- Although the work with input devices is assessed to have negative influence, this perception seems to be very subjective. However, it is essential

to facilitate the usage of VE input devices as well as shutter glasses and cabling.

- Using descriptive text in a Virtual Environment the developers should ensure that the alignment is realized with respect to the user's physical size. Readability should be provided from any point within the CVE interaction space. This is especially interesting when using a CAVE-like display system or a cylindrical projection. In this case descriptive text can be attached to the user's gaze, body or input devices.

- Appropriate tool and input device representations of the remote partner are adequate means for supporting the perception of co-presence which is the basic requirement for collaboration. With the help of these representations the influence of video is reduced to support collaboration only psychologically.

- When integrating immersive telepresence into a CVE, audio and video streams do not necessarily need to be synchronized. Even the resolution plays a tangential role.

6.4.6 Variation Group Analysis

Comparing statistical values of different evaluation sessions as well as comparing group items suffers from the small number of evaluators. Although it is possible to encounter and interpret trends in the answer behaviour the statistical certainties of the computed average values are very low. In order to overcome this problem special evaluation parameters are determined. These parameters change the initial evaluation conditions for different groups. The presumption is that when changing the initial evaluation conditions by the evaluation parameters special answer patterns are provoked. This produces better results than simply evaluating more users under the same conditions. The following evaluation parameters are defined:

1. Remote user's tool and input device representations.

2. Remote user representation using telepresence.

3. Enhanced collaboration.

In the first case the tool and the input device representations of the remote partner are removed from the co-presence and co-work session. This is done in order to force the partners working together to try and substitute the missing representations by other tools as good as possible. This change is assumed to

have a great impact on the perception of co-presence.
The remote user representation (video texture) is removed in another group and it is interesting to see whether the users are still able to work together. Most of the evaluators said that they would have been able to complete the task even without the remote partner's representation as the statistical results in section 6.4.4 indicate.
The last parameter is the collaboration session itself. An enhanced co-work session is chosen in order to force the users to work together in a more collaborative way. The parameter is supposed to have great impact on the perception of co-presence and co-knowledge and thus collaboration. The co-work session is changed in a way that the users do not have to complement the female skeleton by missing bones anymore while cutting skin. The new task is to assemble a snapshot of a running human out of the lying (standing) skeleton by changing the positions of all bones.
The last change in comparison to former evaluations is that the usability questionnaire is handed out together with the co-work questionnaire. The idea is to investigate the influence of the changed evaluation conditions on usability items too.

Statistical Analysis

The abbreviations for the four different evaluation sessions are *Ref* for the reference group, *NT* (no tools) for the group without tool and input device representations, *NV* (no video) for the group without remote user representation and *EC* for the enhanced collaboration task group. Thereby the reference group (*Ref*) is trying to fulfill the same task as the NT and NV group but working with all representations.
The first part of the statistical analysis deals with the investigation of perception of co-presence and the impact on collaboration. The dependencies which are decoded in the group analysis section above are still valid and are considered here again. Hence the perception of co-presence is intended by the following questions (B4, C1, B1, B6, B10, D4, D6):

Question B4		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.5	0.51
2. Group (NT)	5.8	0.30
3. Group (NV)	5.8	0.30
4. Group (EC)	**5.3**	**0.76**

Question C1		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.7	0.49
2. Group (NT)	5.6	0.55
3. Group (NV)	5.7	0.47
4. Group (EC)	**5.6**	**0.61**

Question B1		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.0	0.66
2. Group (NT)	4.5	0.78
3. Group (NV)	**1.7**	**1.13**
4. Group (EC)	5.2	0.69

Question B6		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	4.9	1.12
2. Group (NT)	5.1	0.82
3. Group (NV)	**4.0**	**0.35**
4. Group (EC)	5.0	0.84

Question B10		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	2.4	0.97
2. Group (NT)	4.6	0.85
3. Group (NV)	**4.7**	**0.90**
4. Group (EC)	3.6	1.45

Question D4		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.6	0.50
2. Group (NT)	5.3	0.59
3. Group (NV)	5.0	0.59
4. Group (EC)	**4.8**	**0.82**

Question D6		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.3	0.90
2. Group (NT)	4.6	0.85
3. Group (NV)	4.7	0.68
4. Group (EC)	**4.5**	**0.97**

Lowest ratings of the evaluators are highlighted using a bold font. The EC group assessed the questions B4 and C1 as lowest although the differences between the average values of the other groups are quite small. For example the difference to the highest ratings (NT and NV) is only about 0.5, which lies within the statistical uncertainty. In general it is conspicuous that the lowest assessments are given by the groups NV and EC, the group without video representation and enhanced collaboration session respectively. So for example did the NV group assess question B1 with a very low value of $\overline{x} = 1.7 \pm 1.13$. This result is not surprising whereas the low NV assessment is astonishing. The NV group says the partner is less present due to delays in audio transmission. It seems that people who suffer from the absence of a visual representation have a higher esteem for an audio connection as it is the only communication link to the partner. From this it is clear that the NV again assessed the transfer rate as the factor which is able to increase the perception of co-presence. This result is closely followed by the NT group which suffers from the absence of tool and input device representations and thus esteem an intact audio connection. Additional observer evaluations confirm that the enhanced collaboration group had less problems with orientation within the CVE (D4) but needed more time than others to correct mistakes and continue the work (D6). The NT and NV groups gave low assessments to question D4 as well. The reference group has the highest evaluations of D4 and D6.
This leads to the following conclusion: the enhanced collaboration and the absence of the remote partner representation show the greatest impact of the

evaluation parameters in contrast to the reference group without any changes. Absence of representations seems to affect the perception of co-presence. These users try to compensate for this absence by complementing representations. Working together more collaboratively on a more challenging task has positive influence on orientation within the CVE but leads to more handling errors. Further observations showed that the evaluators of the enhanced collaboration group (EC) lost concentration quicker than other evaluators.

The second part of the statistical analysis deals with the investigation of perception of co-knowledge and the impact on collaboration. Again the dependencies which are decoded in the group analysis section above are still valid and are considered here again. Hence the perception of co-knowledge and collaboration is intended by the following questions (C2, C3, C4, C6, C7, C8, C12, D2):

Question C2	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.9	0.31
2. Group (NT)	6.0	0.00
3. Group (NV)	6.0	0.00
4. Group (EC)	**5.8**	**0.46**

Question C3	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.0	1.02
2. Group (NT)	4.2	0.66
3. Group (NV)	**3.9**	**0.72**
4. Group (EC)	5.4	0.5

Question C4	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.1	1.06
2. Group (NT)	4.2	0.78
3. Group (NV)	**3.1**	**0.92**
4. Group (EC)	5.2	0.51

Question C6	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	**4.0**	**0.76**
2. Group (NT)	5.8	0.30
3. Group (NV)	5.3	0.59
4. Group (EC)	5.1	0.53

Question C7	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.6	0.69
2. Group (NT)	**5.2**	**1.22**
3. Group (NV)	5.2	0.79
4. Group (EC)	5.6	0.70

Question C8	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.4	0.97
2. Group (NT)	5.6	0.67
3. Group (NV)	**5.2**	**0.79**
4. Group (EC)	5.3	0.85

Question C12	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.5	0.96
2. Group (NT)	4.2	0.73
3. Group (NV)	**4.1**	**0.63**
4. Group (EC)	5.6	0.84

Question D2	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	3.8	0.89
2. Group (NT)	4.9	0.38
3. Group (NV)	4.4	1.12
4. Group (EC)	**5.5**	**1.12**

Again, most of the lowest assessments are given by the NV group, the one without remote partner representation. The enhanced collaboration EC group

assessed question C2 lower than the other groups. However, with a value of $\overline{x} = 5.8 \pm 0.46$ this statistical average still belongs to the generally high ratings of this question. Therefore question C1 will not be considered for further evaluations. Because of the absence of the partner and his/her input device and tool representations it becomes evident that the evaluators did not always know where the partner was standing (C4) and doing (C3). Therefore C3 and C4 show low average values for the NT and NV groups with NV a bit lower than NT. In order to compensate for the missing representations these groups spoke more frequently to the partner than other groups (C6). The lowest average value is encountered for the reference group. Obviously there was no need to talk much with the partner as they were working with all representations on an easy collaboration task.
Although there was more communication within the NT and NV than in other teams the evaluators did not have the impression to be equal in rights during the collaboration session (C7). Also the partner is not accepted to be equal in rights as C8 indicates. The collaboration was especially satisfying for the enhanced collaboration on the first place and the reference group on the second place (C12). NT and NV suffered again from the missing representations. The enhanced collaboration task which forces the partner to work together more collaboratively has positive influence on the overall satisfaction. The work was perceived as a success. The drawback is that these partners had to debate more before they could come to an arrangement about a particular action (D2). The observers gave low assessments for the NT and NV groups too.
After this analysis the following conclusions are drawn: it is confirmed again that users try to compensate for missing representations with other tools or forms of communication. To force users into a situation where they have to search for alternatives has negative impact on the perception of co-knowledge and user satisfaction.

The third part of the statistical analysis deals with the investigation of the impact on usability with respect to the changed initial evaluation conditions. Lowest ratings are highlighted using a bold font.

Question A1		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	**5.2**	**0.74**
2. Group (NT)	5.7	0.35
3. Group (NV)	5.2	0.57
4. Group (EC)	5.4	0.50

Question A2		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.5	0.68
2. Group (NT)	4.7	0.38
3. Group (NV)	**4.0**	**0.76**
4. Group (EC)	4.5	0.91

Question A3		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.0	0.59
2. Group (NT)	4.1	0.59
3. Group (NV)	**3.9**	**0.77**
4. Group (EC)	4.5	1.02

Question A4		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	4.9	0.47
2. Group (NT)	3.9	0.83
3. Group (NV)	**2.6**	**0.77**
4. Group (EC)	4.2	0.9

Question A6		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.1	0.61
2. Group (NT)	**4.1**	**0.86**
3. Group (NV)	4.3	0.83
4. Group (EC)	4.9	0.76

Question A7		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	4.8	0.57
2. Group (NT)	**4.2**	**0.66**
3. Group (NV)	4.3	0.90
4. Group (EC)	4.7	0.89

Question A8		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	5.2	1.06
2. Group (NT)	**4.3**	**0.59**
3. Group (NV)	4.3	0.77
4. Group (EC)	4.9	0.72

Question D1		
	$\overline{x}$	$\Delta\overline{x}$
1. Group (Ref)	4.3	0.90
2. Group (NT)	5.4	0.30
3. Group (NV)	**5.8**	**0.97**
4. Group (EC)	5.3	0.76

Most of the lowest assessments are given again by the NT and NV groups. Especially the NV assessments of user comfort concerning input devices (A2), stereo shutter glasses (A3) and cabling (A4) are very low. The average values are closely followed by the low assessments of the NT group. Both groups see the user comfort as problematic while working. The low average value of the reference group for question A1 is not fitting into this scheme. A1 is querying information about the responsiveness of the system. Apart from this, the highest assessments are always given by the reference group. Low average values with a quite high statistical certainty are computed for the NT group without remote tool and input device representations for questions A6, A7 and A8. These questions are querying information about tool representations and whether they are intuitive (A6) and about the alignment of tools (A7) and menus (A8). The external observers confirmed that the evaluators on the NV group had to think the longest before they were able to perform a certain action. It is conspicuous that the order of average values from low to high begins with assessments of the NV group, closely followed by the NT and EC groups. Obviously the enhanced collaboration does not affect usability items negatively although they are lower than the assessment of the reference group. These observations enable to draw the following conclusions.

Conclusions and CVE Rating Scheme

With the variation group analysis it is confirmed that the absence of representation forms has a negative impact on usability. It is proved by the statistical

results from the variation group analysis that a missing remote partner representation handicaps the CVE team more than missing remote tool and input device representations. The intensification of a collaborative work session without restrictions in representations shows impact on usability too. Now in conjunction with the conclusions of evaluation analysis sections 6.4.2 and 6.4.4 it is possible to formulate a CVE rating scheme.
This scheme consists of a chain which starts with the most important thing for a CVE, the audio link to the remote partner. Without audio it is impossible to work adequately. The next component is the video representation of the remote partner. Although this representation form is important it is not essential for the completion of the collaborative task. The users are able to compensate for this missing feature with other adequate tools or forms of communication. The third item is the remote tool and input device representation. These representations support completing the collaborative task but they are also not essential. It is proved from the conclusions of the former analysis sections that compensation always performs at the expense of usability or the perception of co-presence and co-knowledge. Users who do not suffer any missing representation features perceive the collaboration in a CVE as most satisfying. If only one feature is missing the users have to compensate for it by adequate other tools and mechanisms. As a consequence the users are unable to concentrate on the task. The compensating tools and mechanisms stress most of the user's senses in a way that these are occupied and overloaded. Therefore the users perceive the usage of equipment, virtual tool and menus as disturbing and confusing. Users who feel supported are rather willing to accept components which are weak in terms of usability.

6.4.7 Advanced Guidelines

Finally it is possible to formulate some further guidelines with the results obtained by the variation group analysis :

- CVE design and realization should consider the CVE rating scheme.
- An audio link to the remote partner(s)/team needs to be more reliable than a video link.
- A synchronization of audio and video streams is not necessary as long as the delay is not bigger than ten frames.
- Appropriate remote tool and input device representations are supportive but with minor importance relative to the video link.

- If appropriate remote tool and input device representations are difficult to realize ensure that equivalent, compensating tools and mechanisms are offered. Action feedback is able to help overcoming this representation drawback.

- Expert heuristic, formative and summative evaluations of the stand-alone Virtual Environment might not be able to identify weaknesses concerning the usability design for a collaborative Virtual Environment. The alignment of virtual tools and menus as well as the usability of input and output device combinations and other equipment should be designed and implemented with respect to CVE evaluation results.

- Work tools and mechanisms should be designed in order to disburden the users senses. High cognitive load, uncomfortable, non-intuitive usability and user fatigue have negative impact on the perception of co-presence and co-knowledge and thus collaboration.

6.4.8 Conclusions

This chapter introduced an evaluation framework for Collaborative Virtual Environments derived from a Human-Computer-Human interaction model. With the help of this evaluation framework 60 non-VE-expert evaluators assessed the CVE in terms of usability and collaborative awareness. For doing so alternating cycles of expert heuristic, formative and summative evaluation are applied. The statistical analysis of the numeric evaluation results are then used for the formulation of CVE implementation guidelines supporting team work. For performing intelligent evaluation, specific questionnaires and evaluation items are designed and a new analysis method (Variation Group Analysis) is developed.

Chapter 7

Conclusions and Future Work

This thesis is a careful investigation of distributed, collaborative interaction between geographically dispersed teams using projection based Virtual Environments.

In the beginning of this work the basics of Virtual Environments (VE) and Collaborative Virtual Environments (CVE) were described. These are different rear-projection based display systems like the CAVE and the Responsive Workbench. In addition, different devices for interacting with the Virtual Environment were presented such as the Cubic Mouse, pen-like six degree-of-freedom locator devices as well as haptic devices (Phantom). Different software toolkits were discussed with respect to distribution support. Avango as GMD's software framework was used to implement the applications developed in this work. The basic concepts of Avango were described briefly in chapter 1 and in more detail in the implementation chapter 5. Thereby Avango's field interface was described as well as the importance of field connections. Additionally it was explained how the distribution mechanism makes use of this field interface in order to handle the database duplication problem.

The related work relevant to this thesis was presented in chapter 2. This includes a detailed discussion of interaction techniques which are applicable in two-dimensional Graphical User Interfaces (GUIs) as well as in three-dimensional Virtual Environments (VEs). However, it was possible to show that a usage of two-dimensional interaction techniques in three-dimensional VEs is non-trivial. Most of the techniques have to be modified to be able to fulfill three-dimensional requirements. Hence, new techniques especially for Virtual and Collaborative Virtual Environments have been developed. A good reference frame for doing this represents body-relative interaction for rear-projection based Virtual Environments. In combination with other techniques such as speech recognition new and intuitive three-dimensional interaction techniques are designable. Additionally, a survey on selected Collaborative Virtual Environments was presented which were chosen due to their use of tele-immersion approaches. The objectives of the CVE applications were introduced and it was shown how interaction in these Collaborative Virtual Environments is designed.

From the basic chapters 1 and 2 as well as during the presentation of the different CVE approaches it became clear that the absence of a classifying framework for collaborative interaction is the most challenging problem when designing VEs and CVEs. Hence, chapter 3 presents the theoretical approach of this work with which designers and programmers are able to analyze user tasks and interaction cycles in VEs. The results of this analysis, describe a User+Need Space (UNS) which determines the appearance of the Collaborative Virtual Environment. Performing the mapping of the UNS to the CVE representation components is elaborated. These representation components are user representation, remote user representation, data model representation and functionality, environment representation, virtual input device representation and virtual tool representation. In order to find the best interaction the low-level makeup of interaction is analysed. Therefore interaction tasks were narrowed down and interaction templates were found. The developed *Awareness-Action-Feedback (AAF)* loops are such interaction templates. These AAF loops provide the possibility to understand and analyze very tiny steps in interactions. With their help it was possible to form complex interactions through the combination of these loops. Together with the analysis of Awareness-Action-Feedback loops for autonomous and collaborative interaction cycles the desired Application+Interaction Space was determined.

An example application was developed and described in detail in chapter 4 using the theoretical approach. A User+Need Space was determined exemplarily. The interface and application specific information was provided according to a task involving two users and using a Responsive Workbench-Responsive Workbench setup as well as a Responsive Workbench-CAVE setup. Starting with a detailed User Task Description all necessary representation components, metaphors, operations and interaction techniques are determined.
With the description of a currently developed CVE combining a Responsive Workbench, a CAVE and a Cylindrical Display is was possible to exemplify the application design process of very complex CVE. The User Task Description for this CVE describes an adventure game based multi-user experience for practising team work.

CVE implementation details were introduced with respect to the used hardware and the software configuration in chapter 5. The hardware configuration was presented for a distributed setup using the two Responsive Workbenches, and for the distributed setup using a Responsive Workbench and CAVE. These two configurations include input devices for interaction, computers for rendering and distribution, and computers for video and audio streaming. Additionally it includes equipment such as shutter glasses, infra-red emitters, cameras as well as microphones and headphones.
In addition to the detailed description of Avango the video conferencing software for Immersive Telepresence was presented. This was done with respect to the mono and stereo video conferencing capabilities. A solution for sending synchronized stereo video frames was introduced as well as a solution for the integration and rendering of the synchronized stereo images in the CVE.

Evaluation Conclusions

The evaluation of the described and implemented CVE application in chapter 6 was producing many different results assessing usability aspects of CVEs and collaborative awareness in these environments. The theoretical evaluation approach for Collaborative Virtual Environments was derived from the developed Human-Computer-Human interaction model presented in chapter 3. With the help of this evaluation framework 60 non-VE-expert evaluators assessed the CVE.
For doing so, alternating cycles of expert heuristic, formative, and summative evaluation were applied. For performing an advanced evaluation, specific questionnaires and evaluation items were designed. The results of the evaluation were analysed separately according to the Expert Heuristic Analysis, the First Level Analysis, the Group Analysis and the especially developed Variation Group Analysis. The results of these analyses were used for the formulation of CVE design guidelines supporting team work. The usability findings and recommendations of the expert evaluators are concerned with the following items:

- positioning of the toolbar, which groups together generic operations
- handling of the ring menu grouping content specific operations
- tool representations on menus
- three button input device and stylus vs. pinch gloves
- ego-centric vs. exo-centric viewpoint manipulation
- graphical representation of data sets

- video frame rate

From the First Level Analysis, the Group Analysis and the Variation Group Analysis it was possible to present results that concern immersive telepresence, representation of input devices and disturbance factors of collaborative awareness.

First Level Analysis

The results found during the First Level Analysis are:

Immersive Telepresence (First Level Analysis)

- In an educational scenario immersive telepresence supports the work flow. In this situation network drop outs do not have a negative impact on the perception of co-presence as long as the average frame rate does not go below 12fps.

- In a collaboration scenario using immersive telepresence the position of the remote partner representation should be chosen in a way that both partners seem to have same virtual size in the CVE independent from their physical size in real world. This is particularly important when partners are given equal rights for manipulating the data as it was the case in the co-work sessions.

- When using a Responsive Workbench the perception of co-presence can be increased with a remote partner's video texture representation together with a real background. Because of the depth perception the user has the impression that the remote partner stands closer to the table. This effect is not obtainable if the partner's outline is cut out using chroma keying techniques.

Input Device Representations (First Level Analysis)

- Appropriate representations of the remote user's tools and input devices support collaboration more than body and hand gestures.

Disturbance Factors (First Level Analysis)

- Cabling of input devices, trackers and stereo glasses are perceived as annoying. Careful handling of loose wires is recommended.

Group Analysis

For handling the problem of the spread answer patterns group analysis was used in the third part of the analysis. Results obtained by this analysis method concerning immersive telepresence, representation of input devices and disturbance factors are:

Immersive Telepresence (Group Analysis)

- When integrating immersive telepresence into a CVE, audio and video streams do not necessarily need to be synchronized unless the delay is bigger than 10 frames. Even the resolution plays a tangential role, since participants spent most of the time looking and working on the virtual data.

- The experiments show that the remote partner representation is crucial in situations where problems need to be resolved. The use of the video connection enhances the collaboration at a psychological level but its quality can be traded off against other representation components.

Input Device Representations (Group Analysis)

- Appropriate tool and input device representations of the remote partner are adequate means for supporting the perception of co-presence which is the basic requirement for collaboration. With the help of these representations the influence of video is reduced to support collaboration only psychologically.

Disturbance Factors (Group Analysis)

- High system responsiveness is perceived as having very positive impact on collaboration. Even downsizing the application in order to decrease the CPU load is recommendable. A good system responsiveness is guaranteed if all inputs and outputs are processed and rendered within less than 50ms.

- Although the work with input devices is assessed to have negative influence, this perception seems to be very subjective. However, it is essential to facilitate the usage of VE input devices.

- Using descriptive text in a Virtual Environment the developers should ensure that the alignment is realized with respect to the user's physical size. Readability should be provided from any point within the CVE interaction space. This is especially interesting when using a CAVE-like display system or a cylindrical projection. In this case descriptive text can be attached to the user's gaze, body or input devices.

Variation Group Analysis

Although it was possible to encounter and interpret trends in the answer behaviour the statistical certainties of the computed average values were very low. In order to overcome this problem special evaluation parameters were determined. These parameters were changing the initial evaluation conditions for different groups. The presumption was that when changing the initial evaluation conditions by the evaluation parameters special answer patterns were provoked. This was producing better results than simply evaluating more users under the same conditions. The results of this especially developed Variation Group Analysis are:

- With the variation group analysis it is confirmed that the absence of representation forms has a negative impact on usability. It is proved by the statistical results from the variation group analysis that a missing remote partner representation handicaps the CVE team more than missing remote tool and input device representations. The intensification of a collaborative work session without restrictions in representations shows impact on usability too.

- In conjunction with the evaluation results obtained by the former analyses it is possible to formulate a CVE rating scheme.

CVE rating scheme

The CVE rating scheme consists of a chain starting with the audio link to the remote partner, which is proved to be the most important for a Collaborative Virtual Environment. Without audio it is impossible to work adequately. The next component is the video representation of the remote partner. Although this representation form is important it is not essential for the completion of the collaborative task. The users are able to compensate for this missing feature with other adequate tools or forms of communication (i.e. remote tool representation and audio). The third item is the remote tool and input device representation. These representations support completing the collaborative task but they are also not essential.

It is proved from the results of the statistical and group analysis that compensation always performs at the expense of usability or the perception of co-presence and co-knowledge. Users who do not suffer any missing representation features perceive the collaboration in a CVE as most satisfying. If only one feature is missing the users have to compensate for it by other adequate tools and mechanisms. As a consequence, the users are unable to concentrate on the task. The compensating tools and mechanisms stress the user's senses in a way that these become overloaded. Therefore, the users perceive the usage of equipment, virtual tools and menus as disturbing and confusing. Users who feel supported are rather willing to accept components, which are weak in terms of usability.

Further Design Guidelines

Finally it is possible to formulate some further guidelines with the results obtained by the variation group analysis:

- CVE design and realization should consider the CVE rating scheme.
- An audio link to the remote partner(s)/team needs to be more reliable than a video link. Synchronization of audio and video streams is not necessary as long as the delay is not bigger than ten frames.
- Appropriate remote tool and input device representations are supportive but with minor importance relative to the video link. If appropriate remote tool and input device representations are difficult to realize ensure that equivalent, compensating tools and mechanisms are offered. Action feedback is an appropriate solution for overcoming this representation drawback.
- Expert heuristic, formative and summative evaluations of the stand-alone Virtual Environment might not be able to identify weaknesses concerning the usability design for a Collaborative Virtual Environment. The alignment of virtual tools and menus as well as the usability of input and output device combinations and other equipment should be designed and implemented with respect to CVE evaluation results.
- Work tools and mechanisms should be designed for disburdening the users senses. High cognitive load, uncomfortable, non-intuitive usability and user fatigue have negative impact on the perception of co-presence and co-knowledge and thus collaboration.

Future Research

Beside interesting results concerning usability and collaborative awareness in CVEs the thesis offers a great amount of possibilities for further interesting and challenging investigations. The ideas for these further investigations concern the following most important items:

- **Framework:**
  The structure of the taxonomy in chapter 3 is such that additional components and operations and metaphors can be easily incorporated. Refinements of the taxonomy allow for performing better design analyses and for obtaining more subtle results concerning usability evaluations.

- **Persistence:**
  Persistence was not the focus of the thesis. However, it is the basis for long term collaborative work. Persistence here denotes the ability to save and maintain the state of the environment. For the realization of CVEs able to support distance education persistence should be considered. Thus its influence on usability and collaborative awareness should be analysed.

- **Disturbance Factors:**
  It is necessary to investigate further evaluation parameters in order to screen a wider range of disturbance factors that might affect collaborative interaction in CVEs. The more disturbance factors are encountered the more valuable are the evaluation results.

- **Evaluators:**
  Although the Variation Group Analysis is able to reduce the problem of high uncertainty values of the evaluator's answer behaviour, a higher number of experimental subjects should be evaluating the CVE applications. The higher the number of evaluators the smaller the standard deviation and the more certain the evaluation results.

- **Questionnaires:**
  As it was possible to see from the variation group analysis it is necessary to design the questionnaires with respect to the type of analysis to be performed. Future work should try to design refined questionnaires for obtaining further results.

- **Interaction Techniques:**
  This work was not analysing the usability of interaction techniques in particular due to the strong dependency of the task and application. Evaluation of interaction techniques and metaphors, however, needs to be performed for assessing usability problems of these environments.

- **Task Complexity:**
  The implementation of more complex tasks should be considered. This allows for analysing the influence of the task's complexity on the user's behaviour working in the CVE. From the results obtained so far, it will

definitely have an impact on the perception of co-presence and the usability of input devices and tools.

- **Peripheral Influences:**
  For performing advanced usability studies of CVEs peripheral influences such as user preferences should be analyzed. These could consider the influence of highlighting colors, situation dependency, different representations of the remote partner, and positioning of the remote partner.

- **Multiple Sites:**
  The focus of this thesis was on teams working at two different sites. Most probably concepts for team work including more than only two sites will differ. To support multiple sites becomes extremely interesting when investigating the possibilities of CVEs for advanced remote education and game parks. The basis for investigating this special multi-site interaction problem is developed combining the Responsive Workbench, the CAVE and the Cylindrical Display for the adventure game based multi-user experience described in chapter 4.

The list above is a set of suggestions future investigations could go. Nevertheless, with the help of the methodology and tools developed and implemented in this thesis, the basis was created for investigating eminently interesting and diverse problems which are related to the design and development of Collaborative Virtual Environments.
Besides specific results, I hope it was possible to show that Collaborative Virtual Environments represent intriguingly interesting and modern working environments. Their further development as well as their application is extremely challenging and important for our modern society.

Appendix A

Glossary

It is important to define each of the major terms that relate to this work, so that the problem is well understood when talking about collaborative interaction in Virtual Environments. In addition, this list is used by the non-VE expert evaluators as most of the terms appear in the questionnaires (Appendices C-G).

- **display systems** ⇒ Monitors denote display systems for example. There exist even more advanced displays systems, like a room displaying images on each of its four or five walls. This room is called CAVE.
- **stereo glasses, shutter glasses** ⇒ A special type of glasses that enable viewing of artificial stereo images in a Virtual Environment.
- **Virtual Environment, VE** ⇒ VEs are synthetic, computer generated environments which immerse the user and generate the illusion to be at another place.
- **Collaborative Virtual Environment, CVE** ⇒ CVEs are multi-party VEs which allow a number of users to share a common virtual space, where they may interact with each other and the environment itself.
- **being present** ⇒ Denotes the feeling of being in another world or of someone else being in your world.
- **lighting uniformity** ⇒ Lighting uniformity denotes the overall result of all light sources in your environment. They determine the brightness and contrast of most of the things there. The uniformity will determine whether there are parts really bright while others are dark.
- **transfer rate** ⇒ The transfer rate denotes the rate with which your environment imports and exports data from and to other environments.

The higher this transfer rate the more smooth movements of things in your environment appear to you.

- **cabling** ⇒ As most of the technical equipment needs electricity they are connected to energy sources using cables. Additionally cables are used for transferring data from external devices to the computer and vice versa.

- **physical input device** ⇒ Pen like devices or computer game joysticks are electronic input tools for interacting with the virtual world. These physical input devices enable the user to interact with the computer and are often denoted as man-machine-interfaces.

- **tool** ⇒ Tools are representing different operations, actions and functionality. A tool could be a virtual pair of pliers or a hammer or a scalpel for example. They are often presented on special toolbars and menus.

- **virtual tool representation** ⇒ They represent the selected tools which are then attached to the physical input device. This could be a small three-dimensional hammer model following the movements of the user's hand.

Appendix B

Stereo Video Scheme Code

```
;; instantiate and register the DivoService
;; ----------------------------------------------
(define divo-service (make-instance-by-name "fpDivoService"))
(-> divo-service 'register-service "DivoService")

;; instantiate and register the movie service
;; ------------------------------------------
(define movie-service (make-instance-by-name "fpMovieService"))
(-> movie-service 'register-service "MovieService")

;; instantiate a movie texture ;;
---------------------------------------------------
(define movie (make-instance-by-name "fpMovieTex"))

;; instantiate the fpDrawEyes node and
;; configure them for the left and right eye
;; --------------------------------------------------
(define left-eye  (make-instance-by-name "fpDrawEyes"))
(define right-eye (make-instance-by-name "fpDrawEyes"))

;; set both fpDrawEyes nodes active but assign
;; the start to the right eye
;; ---------------------------------------------
(-> (-> left-eye  'Enable) 'set-value 1)
(-> (-> right-eye 'Enable  'set-value 1)
(-> (-> right-eye 'WhichEye 1)

;; instantiate fpFile nodes and
;; load the screen geometry to play the movies on
```

```
;; -------------------------------------------------
(define screen-eye-left  (make-instance-by-name "fpFile"))
(define screen-eye-right (make-instance-by-name "fpFile"))

(-> (-> screen-eye-left  'Filename) 'set-value "./data/video-screenLeft.iv")
(-> (-> screen-eye-right 'Filename) 'set-value "./data/video-screenRight.iv")

;; instantiate a fpDCS for grouping everything
;; and plug the tree together
;; ----------------------
(define screen-dcs (make-instance-by-name "fpDCS"))

(-> (-> left-eye   'Children) 'add-1value EyeLeft)
(-> (-> right-eye  'Children) 'add-1value EyeRight)
(-> (-> screen-dcs 'Children) 'set-value  (list right-eye left-eye))
(-> (-> scene-root 'Children) 'set-value  (list movie screen-dcs))

;; create a material, set it blank, it is later used
;; for overriding the screen geometry
;; ------------------------------------------------
(define material (make-instance-by-name "fpMaterial"))
(-> (-> material 'Diffuse)        'set-value (make-vec3 0 0 0))
(-> (-> material 'Alpha)          'set-value 0.5)
(-> (-> material 'FrontColorMode) 'set-value 0)

;; assign the material to the fpDrawEyes nodes
;; it is used for overriding the material of the screen geometry
;; in the pre- and post-draw-call-backs
;; ------------------------------------------------
(-> (-> left-eye  'Material) 'set-value material)
(-> (-> right-eye 'Material) 'set-value material)

;; determine where the texture comes from and
;; onto which geometry the texture has to be mapped
;; --------------------------------------------------------
(-> (-> movie 'URL)        'set-value  "Divo1")
(-> (-> movie 'NodeList)   'add-1value screen-eye-left)
(-> (-> movie 'NodeList)   'add-1value screen-eye-right)

;; finally play the video ->
;; that means set the download of video texture active
;; ------------------------------------------------
(-> (-> movie 'Playing) 'set-value 1)
```

Appendix C

Introduction questionnaire

- Ihr Alter
  Your age
- Welchen Beruf üben Sie aus ?
  What is your profession ?
- Sind Sie
  Are you
  - ◯ Rechtshänder / right-hander
  - ◯ Linkshänder / left-hander
  - ◯ beidhändig / both-handed
- Wieviele Stunden in der Woche benutzen Sie einen Computer ?
  How often do you use a computer per week ?
  - ◯ weniger als eine Stunde / less than one hour
  - ◯ 1-5 Stunden / 1-5 hours
  - ◯ 5-10 Stunden / 5-10 hours
  - ◯ 10-30 Stunden / 10-30 hours
  - ◯ mehr als 30 Stunden / more than 30 hours
- Mit welchen Eingabegeräten haben Sie Erfahrungen ?
  Which of the following input devices have you used already ?
  - ◯ Mouse
  - ◯ Joystick
  - ◯ Touch Screen

- ○ 3D Eingabegeräte / 3D input devices
- ○ Andere / others :

- Haben Sie jemals Virtuelle Umgebungen benutzt ?
  Have you ever used virtual environments ?
  - ○ häufig / often
  - ○ manchmal / sometimes
  - ○ niemals / never

- Wofür benutzen Sie Ihren Computer ? (Mehrfachnennung möglich)
  What do you use your computer for ? (more than one tick possible)
  - ○ Textverarbeitung / Text editing
  - ○ Spiele / Games
  - ○ Internet surfing
  - ○ 3D Modellierung / 3D Modelling

- Welche der Displays haben Sie schon benutzt ?
  Which of these display systems have you used already ?
  - ○ Monitor (mit/with Mouse und/and Joystick)
  - ○ stereoskopisch/stereoscopic Monitor
  - ○ Stereoskopische Projektionswand / stereoscopic projection Wall
  - ○ Responsive Workbench (TM)
  - ○ CAVE(TM)
  - ○ Head-Mounted Displays

Appendix D

Usability questionnaire

- A1. Hatten Sie das Gefühl, daß ihr VE auf Ihre Eingaben reagiert hat ?
  Did you feel the VE was responding to your actions ?
  - ○ 6 oft/often
  - ○ 5
  - ○ 4
  - ○ 3 manchmal/sometimes
  - ○ 2
  - ○ 1
  - ○ 0 niemals/never
- A2. War das Arbeiten mit den Eingabegeräten komfortabel ?
  Did you feel comfortable working with the input devices ?
  - ○ 6 gut/good
  - ○ 5
  - ○ 4
  - ○ 3 akzeptabel/acceptable
  - ○ 2
  - ○ 1
  - ○ 0 schlecht/bad
- A3. War das Arbeiten mit der Stereobrille komfortabel ?
  Did you feel comfortable working with the stereo glasses ?

- ◯ 6 gut/good
- ◯ 5
- ◯ 4
- ◯ 3 akzeptabel/acceptable
- ◯ 2
- ◯ 1
- ◯ 0 schlecht/bad

• A4. War das arbeiten mit der Verkabelung komfortabel ?
Did you feel comfortable working with the cabling ?

- ◯ 6 gut/good
- ◯ 5
- ◯ 4
- ◯ 3 akzeptabel/acceptable
- ◯ 2
- ◯ 1
- ◯ 0 schlecht/bad

• A5. Wie beurteilen Sie die Darstellung von Schrift innerhalb des VE ?
How do you rate the appearance of text in the VE ?

- ◯ 6 gut/good
- ◯ 5
- ◯ 4
- ◯ 3 akzeptabel/acceptable
- ◯ 2
- ◯ 1
- ◯ 0 schlecht/bad

• A6. Fanden Sie die virtuellen Werkzeugrepräsentationen intuitiv ?
Did you find the virtual tool representations intuitive ?

- ◯ 6 ja, gut/yes, good
- ◯ 5

- ◯ 4
- ◯ 3 akzeptabel/acceptable
- ◯ 2
- ◯ 1
- ◯ 0 nein, schlecht/no, bad

- Wie fanden Sie die Anordnung der
  How do you rate the alignment of the

A7. Werkzeuge ? / tools ?

- ◯ 6 gut/good
- ◯ 5
- ◯ 4
- ◯ 3 akzeptabel/acceptable
- ◯ 2
- ◯ 1
- ◯ 0 schlecht/bad

A8. Menüs ? / menus ?

- ◯ 6 gut/good
- ◯ 5
- ◯ 4
- ◯ 3 akzeptabel/acceptable
- ◯ 2
- ◯ 1
- ◯ 0 schlecht/bad

Appendix E

Co-presence questionnaire

- B1. Hatten Sie das Gefühl, Ihr Partner stehe auf der anderen Seite des Tisches ?
  Did you have the impression your partner was standing on the other side of the table ?
  - ○ 6 ja/yes
  - ○ 5
  - ○ 4
  - ○ 3 manchmal/sometimes
  - ○ 2
  - ○ 1
  - ○ 0 nein/no
- B2. Hatten Sie das Gefühl, daß Ihre Stereobrille und die Ihres Partners ihre Komunikation behindert ?
  Did yours and your partner's stereo glasses hinter your communication ?
  - ○ 6 überhaupt nicht/not at all
  - ○ 5
  - ○ 4
  - ○ 3 manchmal/sometimes
  - ○ 2
  - ○ 1
  - ○ 0 sehr/very

- B3. Hatten Sie den Eindruck, dass Ihr Partner Sie anschaute ?
  Did you have the impression your partner was looking at you ?
  - ◯ 6 ja/yes
  - ◯ 5
  - ◯ 4
  - ◯ 3 manchmal/sometimes
  - ◯ 2
  - ◯ 1
  - ◯ 0 nein/no
- B4. Hatten Sie das Gefühl mit einer Person oder mit einer Maschine zu kommunizieren ?
  Did you have the impression to communicate with a person or with a machine ?
  - ◯ 6 Person
  - ◯ 5
  - ◯ 4
  - ◯ 3 künstliche Person / artificial person
  - ◯ 2
  - ◯ 1
  - ◯ 0 Maschine
- B5. Hatten Sie wegen der Zeitverzögerung in der Videoübertragung das Gefühl, daß Ihr Partner weniger präsent wäre ?
  Did you have the impression due to the delay in video transmission that your partner was less present ?
  - ◯ 6 nein/no
  - ◯ 5
  - ◯ 4
  - ◯ 3 manchmal/sometimes
  - ◯ 2
  - ◯ 1
  - ◯ 0 ja/yes

- B6. Hatten Sie wegen der Zeitverzögerung in der Audioübertragung das Gefühl, daß Ihr Partner weniger präsent wäre ?
  Did you have the impression due to the delay in audio transmission that your partner was less present ?
  - ◯ 6 nein/no
  - ◯ 5
  - ◯ 4
  - ◯ 3 manchmal/sometimes
  - ◯ 2
  - ◯ 1
  - ◯ 0 ja/yes
- B7. Sind Sie der Meinung, daß das Videobild Ihres Partner wirklich notwendig ist ?
  Do you think that the video image of your partner is really necessary ?
  - ◯ 6 ja/yes
  - ◯ 5
  - ◯ 4
  - ◯ 3 manchmal/sometimes
  - ◯ 2
  - ◯ 1
  - ◯ 0 nein/no
- B8. Sind Sie der Meinung, daß Sie Ihre Aufgaben ebenfalls ohne das Videobild Ihres Partner vollenden könnten ?
  Do you think you could complete your task even without the video image of your partner ?
  - ◯ 6 ja/yes
  - ◯ 5
  - ◯ 4
  - ◯ 3 vielleicht/maybe
  - ◯ 2
  - ◯ 1

- ◯ 0 nein/no

- B' Welchen Hintergrund hinter Ihrem Partner haben Sie bevorzugt ?
  Which background behind your partner did you prefer ?

  - ◯ Kamerahintergrund / camera background
  - ◯ künstlicher Hintergrund (z. B. Operationsraum) / artificial background (e.g. operating room)
  - ◯ uniformer Hintergrund (z. B. blauer, schwarzer Vorhang) / uniform background (e.g. blue, black curtain)

- Welche der folgenden Faktoren, würden die Präsenz Ihres Partners für Sie erhöhen ?
  Which of the following factors can increase the co-presence of your partner ?

  B9. Ausleuchtung / lighting uniformity

  - ◯ 6 sehr/very
  - ◯ 5
  - ◯ 4
  - ◯ 3 vielleicht/maybe
  - ◯ 2
  - ◯ 1
  - ◯ 0 ueberhaupt nicht/not at all

  B10. Übertragungsgeschwindigkeit / transfer rate

  - ◯ 6 sehr/very
  - ◯ 5
  - ◯ 4
  - ◯ 3 vielleicht/maybe
  - ◯ 2
  - ◯ 1
  - ◯ 0 ueberhaupt nicht/not at all

  B11. Position und Größe des Partners / partners position and size

- ◯ 6 sehr/very
- ◯ 5
- ◯ 4
- ◯ 3 vielleicht/maybe
- ◯ 2
- ◯ 1
- ◯ 0 überhaupt nicht/not at all

Appendix F

Co-work questionnaire

- C1. Hatten Sie das Gefühl mit einer Person oder mit einer Maschine zu kommunizieren ?
  Did you have the impression to communicate with a person or a machine ?
  - ◯ 6 Person
  - ◯ 5
  - ◯ 4
  - ◯ 3 künstliche Person / artificial person
  - ◯ 2
  - ◯ 1
  - ◯ 0 Maschine
- C2. Würden Sie nochmal mit demselben Partner zusammenarbeiten ?
  Would you collaborate with the same partner again ?
  - ◯ 6 ja/yes
  - ◯ 5
  - ◯ 4
  - ◯ 3 vielleicht/maybe
  - ◯ 2
  - ◯ 1
  - ◯ 0 nein/no
- C3. Wußten Sie immer, was Ihr Partner gerade tut ?
  Did you always know what your partner was doing ?

- ◯ 6 ja, immer/yes, always
- ◯ 5
- ◯ 4
- ◯ 3 manchmal/sometimes
- ◯ 2
- ◯ 1
- ◯ 0 nein, nie/no, never

• C4. Wußten Sie immer, wo Ihr Partner gerade steht ?
Did you always know where your partner was standing ?

- ◯ 6 ja, immer/yes, always
- ◯ 5
- ◯ 4
- ◯ 3 manchmal/sometimes
- ◯ 2
- ◯ 1
- ◯ 0 nein, nie/no, never

• C5. Wie oft haben Sie Ihren Partner angeschaut ?
How often did you look at your partner ?

- ◯ 6 häufig/often
- ◯ 5
- ◯ 4
- ◯ 3 manchmal/sometimes
- ◯ 2
- ◯ 1
- ◯ 0 nie/never

• C6. Wie oft haben Sie zu Ihrem Partner gesprochen ?
How often did you speak to your partner ?

- ◯ 6 häufig/often
- ◯ 5

- ○ 4
- ○ 3 manchmal/sometimes
- ○ 2
- ○ 1
- ○ 0 nie/never

- C7. Hatten Sie das Gefühl während der Ausführung Ihrer Aufgabe gleichberechtigt zu sein ?
  Did you have the impression you had an equal part while working together ?

  - ○ 6 ja/yes
  - ○ 5
  - ○ 4
  - ○ 3 mehr oder weniger/more or less
  - ○ 2
  - ○ 1
  - ○ 0 nein/no

- C8. Sahen Sie Ihren Partner als gleichberechtigt an ?
  Did you see your partner as equal ?

  - ○ 6 ja/yes
  - ○ 5
  - ○ 4
  - ○ 3 mehr oder weniger/more or less
  - ○ 2
  - ○ 1
  - ○ 0 nein/no

- C9. Haben die Hand und Körpergesten Ihres Partners die Zusammenarbeit unterstützt ?
  Did hand and body gestures of your partner support your collaboration ?

  - ○ 6 ja/yes

- ◯ 5
- ◯ 4
- ◯ 3 mehr oder weniger / more or less
- ◯ 2
- ◯ 1
- ◯ 0 nein/no

- C10. Sind Sie der Meinung, daß das Videobild Ihres Partner wirklich notwendig ist ?
  Do you think that the video image of your partner is really necessary ?
  - ◯ 6 ja/yes
  - ◯ 5
  - ◯ 4
  - ◯ 3 manchmal/sometimes
  - ◯ 2
  - ◯ 1
  - ◯ 0 nein/no
- C11. Sind Sie der Meinung, daß Sie Ihre Aufgaben ebenfalls ohne das Videobild Ihres Partner vollenden könnten ?
  Do you think you could complete your task even without the video image of your partner ?
  - ◯ 6 ja/yes
  - ◯ 5
  - ◯ 4
  - ◯ 3 vielleicht/maybe
  - ◯ 2
  - ◯ 1
  - ◯ 0 nein/no
- C12. War die gesamte Zusammenarbeit befriedigend ?
  Was the overall collaboration satisfying ?
  - ◯ 6 ja/yes

- ○ 5
- ○ 4
- ○ 3 mehr oder weniger / more or less
- ○ 2
- ○ 1
- ○ 0 nein/no

Appendix G

Observer questionnaire

- D1. Wie lange mußte der Benuzter überlegen, um eine Tätigkeit auszuführen ?
  How long did the user have to think before performing an action ?
  - ◯ 6 kurz/shortly
  - ◯ 5
  - ◯ 4
  - ◯ 3
  - ◯ 2
  - ◯ 1
  - ◯ 0 lange/long
- D2. Mußten sich die Benutzer lange besprechen eine Tätigkeit auszuführen ?
  Did the users have to debate a lot before performing an action ?
  - ◯ 6 ja/yes
  - ◯ 5
  - ◯ 4
  - ◯ 3 mehr oder weniger / more or less
  - ◯ 2
  - ◯ 1
  - ◯ 0 nein/no

- D3. Hatte der Benutzer Probleme Werkzeuge mit seiner/ihrer rechten oder linken hand auszuwählen ?
  Did the user have problems to select tools with his/her right or left hand ?
  - ◯ 6 ja/yes
  - ◯ 5
  - ◯ 4
  - ◯ 3 mehr oder weniger / more or less
  - ◯ 2
  - ◯ 1
  - ◯ 0 nein/no
- D4. Hatte der Benutzer Probleme mit seiner/ihrer Orientierung ?
  Did the user have problems with orientation ?
  - ◯ 6 ja/yes
  - ◯ 5
  - ◯ 4
  - ◯ 3 mehr oder weniger / more or less
  - ◯ 2
  - ◯ 1
  - ◯ 0 nein/no
- D5. War es offensichtlich, daß der Benutzer während der Session an Konzentration verlor ?
  Did the user obviously loose concentration during a session ?
  - ◯ 6 ja/yes
  - ◯ 5
  - ◯ 4
  - ◯ 3 nicht signifikant / not significant
  - ◯ 2
  - ◯ 1
  - ◯ 0 nein/no

- D6. Wie schnell konnte der Benutzer Fehler korrigieren und die Arbeit fortsetzen ?
  How quickly could the user correct mistakes and continue the work ?
  - ◯ 6 sehr schnell/very quickly
  - ◯ 5
  - ◯ 4
  - ◯ 3
  - ◯ 2
  - ◯ 1
  - ◯ 0 sehr langsam/very slowly
- D7. Hatte der Benutzer Spaß während der Session ?
  Did the user have fun during the session ?
  - ◯ Ja / Yes
  - ◯ Nein /No
- Informal observations of the user:

Bibliography

[1] I. Angus and H. Sowizral. Embedding the 2d interaction metaphor in a real 3d virtual environment. In *Stereoscopic Displays and Virtual Reality Systems, 2409*, pages 282–293, SPIE, 1995.

[2] R. Beaten and R. DeHoff. An evaluation of input devices for 3-d computer display workstations. In *The International Society for Optical Engineering*, pages 237–244, SPIE, 1987.

[3] S. Benford, J. Bowers, L.E. Fahlén, J. Mariani, and T. Rodden. Supporting co-operative work in virtual environments. *The Computer Journal*, 37(8), 1994.

[4] S. D. Benford, C. C. Brown, G. T. Reynard, and C. M. Greenhalgh. Shared spaces: Transportation, artificiality and spatiality. *Proc. ACM Conference on Computer Supported Cooperative Work (CSCW'96), ACM Press*, 1996.

[5] A. Bierbaum. *VR Juggler: A Virtual Platform for Virtual Reality Application Development*. Master Thesis, Iowa State University, 2000.

[6] S.A. Bly, S.R. Harrison, and S. Irwin. Media spaces: Video, audio, and computing. *Communications of the ACM*, 36(1):28–47, 1993.

[7] K.R. Boff, L. Kaufman, and J.P. Thomas (Eds.). *Handbook of Perception and Human Performance*. New York, John Wiley and Sons, 1986.

[8] U.M. Borghoff and J.H. Schlichter. *Rechnerunterstützte Gruppenarbeit*. Springer, 1998.

[9] D. Bowman. Interaction techniques for immersive virtual environments: Design, evaluation, and application. *Human-Computer Interaction Consortium (HCIC) Conference*, 1998.

[10] D. Bowman, E. Davis, A. Badre, and L. Hodges. Maintaining spatial orientation during travel in an immersive virtual environment. *Presence: Teleoperators and Virtual Environments*, 8(6):618–631, 1999.

[11] D. Bowman and L. Hodges. Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments. *The Journal of Visual Languages and Computing*, 10(1):37–53, 1999.

[12] D. Bowman and L. Hodges. Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments. *The Journal of Visual Languages and Computing*, 10(1):37–53, 1999.

[13] D. Bowman, L. Hodges, and J. Bolter. The virtual venue: User-computer interaction in information-rich virual environments. *Presence: Teleoperators and Virtual Environments*, pages 478–493, 1998.

[14] D. Bowman, D. Johnson, and L. Hodges. Testbed evaluation of ve interaction techniques. *Proceedings of VRST'99*, pages 26–33, 1999.

[15] D. Bowman, D. Koller, and L. Hodges. Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. *Proceedings of VRAIS'96*, pages 45–52, 1997.

[16] D. Bowman, E. Kruijff, J. LaViola, M. Mine, and I. Poupyrev. Three-dimensional user interfaces: Fundamental techniques, theory, and practice. *Course presented at ACM SIGGRAPH*, 2000.

[17] D. Bowman, E. Kruijff, J. LaViola, and I. Poupyrev. An introduction to 3d user interface design. *Presence: Teleoperators and Virtual Environments*, 10(1):96–108, 2001.

[18] D. Bowman, J. LaViola, I. Poupyrev, and M. Mine. Advanced topics in 3d user interface design. *Course presented at ACM SIGGRAPH*, 2001.

[19] J. Brederson, M. Ikits, C. Johnson, and C. Hansen. The visual haptic workbench, 2000.

[20] C. Breiteneder, S.J. Gibbs, and C. Arapis. Teleport- an augmented reality teleconferencing environment. *Proc. 3rd Eurographics Workshop on Virtual Environments Coexistence and Collaboration*, February 1996.

[21] W. Broll. Interacting in distribute collaborative virtual environments. *IEEE VRAIS*, pages 148–155, 1995.

[22] G. Burdea. *Force and Touch Feedback for Virtual Reality*. Wiley Interscience, 1996.

[23] T. Capin, I. Pandzic, N. Magnenat-Thalmann, and D. Thalmann. *Avatars in Networked Virtual Environments*. John Wiley and Sons, 1999.

[24] C. Carlssson and O. Hagsand. Dive - a platform for multi-user virtual environments. *Computers and Graphics*, 17(6):663–669, Nov.-Dec. 1993.

[25] S. Chatty. Issues and experience in designing two-handed interaction. *ACM CHI*. 1993.

[26] S. Chatty. Extending a graphical toolkit for two-handed interaction. *ACM UIST'94*, 1994.

[27] C. Cruz-Neira, D.J. Sandin, T.A. DeFanti, R. Kenyon, and J.C. Hart. The cave, audio visual experience automatic virtual environment. *Communications of the ACM*, June 1992.

[28] C. M. Curry. *Supporting Collaborative Awareness in Tele-Immersion.* Master Thesis, Virginia Polytechnic Institute and State University, 1999.

[29] R. Darken. Hands-off interaction with menus in virtual space. In *Proceedings of Stereoskopic Displays and Virtual Reality Systems*, pages 365–371, SPIE, 1994.

[30] D. Diaper. *Task Analysis for Human-Computer Interaction.* Ellis Horwood Limited, 1989.

[31] E. Eberleh, H. Oberquelle, and R. Oppermann. *Einführung in die Software-Ergonomie.* de Gruyter, 1994.

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

[33] C. Faisstnauer, D. Schmalstieg, and Z. Szalavari. Device-independent navigation and interaction in virtual environments. *Vienna Universtiy of Technology, Austria*, 1996.

[34] S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon. Windows on the world: 2d windows for 3d augmented reality. *UIST*, pages 145–155, 1993.

[35] W. Felger. *Innovative Interaktionstechniken in der Visualisierung.* Springer Verlag, Berlin Heidelberg, 1995.

[36] S. Fisher, M. McGreevy, J. Humphries, and W. Robinett. Virtual environment display system. In *Proceedings of Workshop on Interactive 3D Graphics*, ACM, 1986.

[37] M. Fjeld, M. Bichsel, and M. Rauterberg. Build-it: a brick-based tool for direct interaction. *In D. Harris (ed.) Engineering Psychology and Cognitive Ergonomics (EPCE), Vol. 4. Hampshire: Ashgate*, pages 205–212, 1999.

[38] J. Foley, A. van Dam, S. Feiner, and J. Hughes. Computer graphics principles and practice, 1990.

[39] J. D. Foley, V. L. Wallace, and P. Chan. The human factors of computer graphics interaction techniques. *IEEE Computer Graphics and Applications*, 4:13–48, 1984.

[40] E. Frecon and A. A. Nou. Building distributed virtual environments to support collaborative work. *VRST*, pages 105 – 113, 1998.

[41] B. Fröhlich, S. Barrass, B. Zehner, J. Plate, and M. Göbel. Exploring GeoScience Data in Virtual Environments. In *Proc. Visualization 99*, 1999.

[42] B. Fröhlich and J. Plate. The Cubic Mouse, A new device for Three-Dimensional Input. In *Proceedings of CHI 2000*, 2000.

[43] A. Fuhrmann, G. Hesina, F. Faure, and M. Gervautz. Occlusion in Collaborative Augmented Environments. In *Proc. of the Eurographics Workshop in Vienna, Austria*, May 31 - June1, 1999.

[44] J. L. Gabbard and D. Hix. A taxonomy of usability characteristics in virtual environments. *Technical report, Virginia Polytechnic Institute and State University, Final Report to the Office of Naval Research.*, 1997.

[45] S.J. Gibbs, C. Arapis, and C. Breiteneder. TELEPORT - towards immersive copresence. *Multimedia Systems*, 7(3):214–221, 1999.

[46] E. Gobetti, J.-F. Balaguer, and D. Thalmann. Vb2:an architecture for interaction in synthetic worlds. *Proc. ACM UIST '93, Atlanta*, pages 167–178, 1993.

[47] G. Goebbels, P. Aquino, V. Lalioti, and M. Göbel. Supporting team work in collaborative virtual environments. *In Proceedings of ICAT 2000 - The Tenth International Conference on Artificial Reality and Tele-existence*, pages 104–111, Oct. 2000.

[48] 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.

[49] G. Goebbels, V. Lalioti, and M. Göbel. On collaboration in distributed virtual environments. *In The Journal of Three Dimensional Images, Japan*, 14(4):42–47, 2000.

[50] G. Goebbels, V. Lalioti, and T. Mack. Guided design and evaluation of distributed, collaborative 3d interaction in projection based virtual environments. *In: Usability Evaluation and Interface Design: Cognitive Engineering, Intelligent Agents and Virtual Reality, Conference Proceedings HCI International 2001, eds. Smith, Salvendy, Hariis, and Koubek*, 1:26–30, 2001.

[51] G. Goebbels and X. Yang. Distributed medical teaching through responsive workbench and cyberstage. *In Proceedings of ICIG2000*, Tianjin, China, Aug. 2000.

[52] R. Harmon, W. Patterson, W. Ribarsky, and J. Bolter. The virtual annotation system. In *Proceedings of VRAIS'96*, pages 239–245, IEEE, 1996.

[53] D. Hix and H. R. Hartson. *User Interface Development: Ensuring Usability through Product and Process.* New York: John Wiley and Sons, 1993.

[54] D. Hix, E. Swan II, J. L. Gabbard, M. McGee, J. Durbin, and T. King. User-centered design and evaluation of a real-time battlefield visualization virtual environment. *IEEE*, pages 96–103, 1999.

[55] Y. Ichikawa, K. Okada, G. Jeong, S. Tanaka, and Y. Matushita. Majic videoconferencing system: Experiments, evaluation and improvement. *In Proceedings of ECSCW95*, 1995.

[56] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A sketching interface for 3d freeform design. *Proceedings of SIGGRAPH 99, Los Angeles, CA*, 1999.

[57] H. Ishii and M. Kobayishi. Integration of inter-personal space and shared workspace: Clearboard design and experiments. *In Proceedings of CSCW'92*, pages 33–42, 1992.

[58] H. Ishii and B. Ullmer. The metadesk: Models and prototypes for tangible user interfaces. *In Proceedings of UIST'97*, pages 223–232, 1997.

[59] R. Jacoby and S. Ellis. Using virtual menus in a virtual environment. In *Proceedings of Visual Data Interpretation*, pages 39–48, SPIE, 1992.

[60] A. Johnson, M. Roussos, J. Leigh, C. Barnes, C. Vasilakis, and T. Moher. The nice project: Learning together in a virtual world. *In Proceedings of VRAIS'98*, 1998.

[61] B. Jung, S. Kopp, M. E. Latoschik, T. Sowa, and I. Wachsmuth. Virtuelles konstruieren mit gestik und sprache. *KI*, pages 5–11, 2000.

[62] M. Kallmann and D. Thalmann. Modeling objects for interaction tasks. *Proc. Eurographics Workshop on Animation and Simulation*, pages 73–86, 1998.

[63] W. Krüger, C. Bohn, B. Fröhlich, H. Schüth, W. Strauss, and G. Wesche. The responsive workbench: A virtual work environment. *IEEE Computer*, pages 12–15, May 1994.

[64] W. Krüger and B. Fröhlich. The responsive workbench. *IEEE Computer Graphics and Applications*, May 1994.

[65] H. Kuchling. *Taschenbuch der Physik*. 12. durchges. Aufl., Verlag Harri Deutsch, Frankfurt/Main, Thun 1989.

[66] H. Kuzuoka, G. Ishimoda, and T. Nishimura. Can the gesturecam be a surrogate? *In Proceedings of ECSCW95*, 1995.

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

[68] J. Liang and M. Green. Jdcad: a highly interactive 3d modeling system. *In Proc. Third International Conference on CAD and Computer Graphics, Beijing, China*, pages 217–222, 1993.

[69] D. Margery, B. Arnaldi, and N. Plouzeau. A General Framework for Cooperative Manipulation in Virtual Environments. In *Proc. of the Eurographics Workshop in Vienna, Austria*, May 31 - June1, 1999.

[70] M. Mine. Moving cows in space: Exploiting proprioception as A framework for virtual envrionment interaction. Technical Report TR97-003, Department of Computer Science, University of North Carolina - Chapel Hill, February 1997.

[71] 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.

[72] M. R. Mine. Virtual environment interaction techniques. Technical Report Technical Report TR95-018, Chapel Hill, Department of Computer Science, University of North Carolina, 1995.

[73] J.D. Mulder, R. van Liere, and J. van Wijk. Computational steering in the cave. *FGCS - Future Generation Computer Systems*, 14(3-4):199-209, August 1998.

[74] J. Nielson. *Usability Engineering*. Academic Press, 1993.

[75] D. A. Norman. *The Psychology of Everyday Things*. Basic Books, New York, 1988.

[76] T. Poston and L. Serra. Dextrous virtual work. In *Communications of the ACM*, volume 39(5), pages 37–45, 1996.

[77] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. Go-go interaction technique: Non-linear mapping for direct manipulation in vr. In *Proceedings of UIST'96*, pages 79–80, ACM, 1996.

[78] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of SIGGRAPH'98*, pages 179–188, ACM, 1998.

[79] M. Rauterberg. Usability evaluation: an empirical validation of different measures to quantify interface attributes. *In: T. Sheridan (ed.), Analysis, Design and Evaluation of Man-Machine Systems*, 2:467–472, Oxford: Pergamon 1995.

[80] M. Roussos, A. Johnson, T. Moher, J. Leigh, C. Vasilakis, and C. Barnes. Learning and building together in an immersive virtual world. *In Presence vol 8, no 3, June, 1999, special issue on Virtual Environments and Learning; edited by William Winn and Michael J. Moshell., MIT Press*, pages 247–263, 1999.

[81] M. Roussos, A.E. Johnson, J. Leigh, Ch. A. Vasilakis, C.R. Barnes, and T.G. Moher. Nice: Combining constructionism, narrative and collaboration in a virtual learning environment. *Computer Graphics*, 31(3):62–63, 1997.

[82] D. Ruspini, K. Kolarov, and O. Khatib. Haptic interaction in virtual environments. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems: IROS'97, September 1997, Grenoble, France.*, 1997.

[83] S. Sellen and B. Buxton. Using spatial cues to improve videoconferencing. *In Proceedings of CHI92*, pages 651–652, 1992.

[84] Corporation Sense8. World2world release1 technical overview. Technical report, Sence8 Corporation, 1997.

[85] Corporation Sense8. Worldtoolkit technical overview. Technical report, Sence8 Corporation, 1998.

[86] SGI. Digital media development environment. Technical report, Silicon Graphics Inc.

[87] SGI. Origin 2000 and onyx 2 performance tuning and optimization guide. Technical report, Silicon Graphics Inc.

[88] SGI. Performer getting started guide. Technical report, Silicon Graphics Inc.

[89] SGI. Performer programmers's guide. Technical report, Silicon Graphics Inc.

[90] M. Slater. Measuring presence: A response to the witmer and singer presence questionnaire, 1999.

[91] M. Slater and A. Steed. A virtual presence counter. *Presence: Teleoperators and Virtual Environments 9.5*, 2000.

[92] A. Steed, M. Slater, A. Sadagic, A. Bullock, and J. Tromp. Leadership and collaboration in shared virtual environments. *IEEE Virtual Reality, Houston*, pages 112–115, March 1999.

[93] R. Stoakley, M. Conway, and R. Pausch. Virtual reality on a wim: Interactive worlds in miniature. In *Conference on Human Factors in Computing Systems*, pages 265–272, 1995.

[94] I. Sutherland. The ultimate display. *Proceedings of IFIP Congress.*, pages 505–508, 1965.

[95] H. Takemura and F. Kishino. Cooperative work environment using virtual workspace. *In Proceedings of CSCW92*, 1992.

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

[97] M. Usoh, E. Catena, S. Arman, and M. Slater. Presence questionnaires in reality. *Presence: Teleoperators and Virtual Environments , in press*, 2000.

[98] R. van Liere and J.D. Mulder. PVR - An Architecture for Portable VR Applications. In *Proc. of the Eurographics Workshop in Vienna, Austria*, May 31 - June1, 1999.

[99] J. Verlinden, J. Bolter, and C. van der Mast. Virtual annotation: Verbal communication in virtual reality. In *Proceedings of European Simulation Symposium*, pages 305-310, SCS Gent, Belgium, 1993.

[100] I. Wachsmuth. Kommunikative rhythmen in gestik und sprache. *Kognitionswissenschaft 8(4)*, pages 151-159, 2000.

[101] I. Wachsmuth, I. Voss, T. Sowa, M. E. Latoschik, and B. Jung. *Multimodale Interaktion in der Virtuellen Realität.* Mensch und Computer 2001, 2001.

[102] Josie Wernecke and Open Inventor Architecture Group. *Inventor Mentor, The: Programming Object-Oriented 3D Graphics with Open Inventor, Release 2.* Addison Wesley, 1994.

[103] B.G. Witmer. and M.J. Singer. Measuring immersion in virtual environments. *(Tech. Report 1014). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences*, 1994.

[104] R. Ziegler. Haptic displays - how can we feel virtual environments? *In Proceedings of Eurographics'96*, France 1996.