Supporting Collaborative Awareness in Tele-immersion

Kevin Michael Curry

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science and Applications

Dr. Mary-Beth Rosson, Chair
Dr. John M. Carroll
Dr. Ronald D. Kriz

June 24, 1999
Blacksburg, Virginia

Keywords: Tele-immersion, CSCW, virtual reality, collaborative awareness, CAVE

Supporting Collaborative Awareness in Tele-immersion

Kevin Michael Curry

(ABSTRACT)

The goal of this thesis is to present the virtual environments research community with a thorough investigation of collaborative awareness in Tele- immersion and related immersive virtual environments. Tele-immersion was originally defined in 1996 by Tom Defanti of the Electronic Visualization Laboratory (EVL), is "the union of networked VR and video in the context of significant computing and data mining" [Leigh, et. al., 1997]. Since then, research on Tele-immersion has outgrown most of its system and performance-related issues and now focuses supporting collaborative interaction and usability. Tele-immersion now deals with the "[creation of persistent virtual environments] enabling multiple, globally situated participants to collaborate over high-speed and high-bandwidth networks connected to heterogeneous supercomputing resources and large data stores" [Leigh, et. al., 1997, p. 1 of 9].

In the early stages of Tele- immersion there were two main factors driving the research: the significant processing load of real-time and simulated computational steering, and the sheer bulk of the data sets being generated for scientific visual analysis [Leigh, et. al., 1997]. Now the growing number of immersive VR sites is motivating a need to support human-to-human interaction and work over wide networks of immersive virtual environments. This research focuses heavily on issues of collaborative awareness in these networked, immersive virtual environments. Collaborative awareness, in this context, is a concept that encompasses the caveats of one's knowledge about the CVE and its occupants. As a result of this study, software has been designed to provide support not only for collaborative awareness, but also for several other dimensions of collaboration.

Dedicated to my dear grandmothers, Anne and Doris

Acknowledgments

While this paper is my own work, many people to whom I wish to express my most sincere thanks supported it. I would like to acknowledge my committee chair Dr. Mary Beth Rosson, who first inspired me to investigate collaboration in the CAVE and who continued to support that investigation throughout the remainder of my graduate career. Also Dr. John M. Carroll, who brought focus to this paper and challenged me to provide the very best treatment I could. And to Dr. Jason Leigh, who not only shared the fruits of his labor in building a programmatic foundation, but also continued to support this work by providing experience, knowledge and opportunity.

I am particularly grateful for the support of the team at VT-CAVE for everything they have contributed to make this work possible. To Kent Swartz, who inspired the earliest inspiration and support for collaborative awareness and who has worked with me on the Console from the beginning. To John Kelso, who played many roles and never once failed to live up to any of them. And to Dr. Ronald D. Kriz whose contributions I can only begin to describe; sponsor, advocate, mentor.

Finally, I would like to thank members of my family. To my father, Pat Curry for providing both encouragement and considerable insight into the world of modeling and simulation in the defense industry. And to my loving wife, Carolyn who has endured my long hours and late nights in front of the computer while remaining steadfastly supportive and enthusiastic toward everything I do.

Preface

The goal of this thesis is to present the virtual environments research community with thorough investigation of collaborative awareness in Tele-immersion and related immersive computing environments.

The first chapter starts with a brief introduction to the concept of Tele-immersion [DeFanti in Leigh, et. al., 1997]. This is meant to help familiarize readers with the nature of the human-computing environment. Given that collaborative nature of interaction in Tele-immersion, a connection with Computer-Supported Cooperative Work (CSCW) is then made. This will serve to provide support for new postulations when research in Tele-immersion, and collaborative virtual environments (CVEs) in general, fails to address our needs. The remainder of the first chapter establishes the scope and approach taken by this research.

The second chapter of this paper sets up the background and the motivations behind the research. It begins with a definition of Tele-immersion. Motivations are linked to the growing number of immersive projection VR facilities and the desire to link them together into a robust collaborative network. Some examples of how these systems are, and can be used are provided. These examples are followed by summary descriptions of some hardware and software configurations for implementing Tele-immersion. At this point readers should have enough preliminary information to start thinking about what might be possible with Tele-immersion. To put some of these ideas into context, the chapter ends with several scenarios of actual and proposed use. Claims analysis is used to bring out the major points in each scenario.

The third chapter deals entirely with the concept of collaborative awareness. Many relevant issues from CSCW were considered for this treatment. Elements of these issues, particularly with regard to collaborative awareness, have emerged in CVEs and even in Tele-immersion. These elements are organized and covered in some detail. In particular, this information should provide some foundation that will allow us to extend support for collaborative *interaction* in Tele-immersion.

The fourth chapter makes a transition into applying what has been learned about collaborative awareness in Tele-immersion to the design and implementation of a collaborative application for Tele-immersion. In this chapter a prototype, called the CAVE Collaborative Console, is covered in detail. The Console is a software interface that enables collaborative interaction over Tele-immersive networks. Each component and feature of the prototype is covered separately.

The fifth chapter summarizes the research contributions and addresses topics for future work. During the course of the research, many interesting observations were made which could not be given full attention and treatment. These are listed in brief detail in hope of motivating continues work in this area.

Chapter One:
Introduction to Research

Tele-immersion was originally defined in 1996 by Tom Defanti of the Electronic Visualization Laboratory (EVL), is “the union of networked VR and video in the context of significant computing and data mining” [Leigh, et. al., 1997]. Since then, research onTele-immersion has outgrown most of its system and performance-related issues now focuses supporting collaborative interaction and usability. Tele-immersion now deals with the "[creation of persistent virtual environments] enabling multiple, globally-situated participants to collaborate over high-speed and high-bandwidth networks connected to heterogeneous supercomputing resources and large data stores” [Leigh, et. al., 1997, p. 1 of 9]. In the early stages of Tele-immersion there were two main factors driving the research: the significant processing load of real-time and simulated computational steering, and the sheer bulk of the data sets being generated for scientific visual analysis [Leigh, et. al., 1997]. Now the growing number of immersive VR sites is motivating a need to support human-to-human interaction and *work* over wide networks of immersive virtual environments.

Today over 100 installations worldwide provide a rich platform for Tele-immersion. These facilities are equipped with a variety of immersive projection technologies, including: CAVE Automated Virtual Environments (CAVEs), Immersa-Desks (I-Desks, I-Walls), and CAVE Simulators [URL 1]. There are even more facilities investigating or using other types of immersive projection technology (IPT) which are not listed with NCSA. Both user populations are growing. With the growth in the number of immersive VR sites comes an expected growth in the number of sites that will connect to one another over electronic networks. This connection is not only inspired by a desire to exchange data and resources, but more importantly, to *share* them in real time as part of a collaborative virtual environment (CVE). Already, there is an international “users’ group” called the CAVE Research Network Users’ Society (CAVERNUS). As more people engage in immersive, collaborative interaction, human factors become more important and we must consider many new issues. These new issues are just beginning to be perceived, understood and addressed. Much of what is currently known about human factors in Tele-immersion speaks to issues of performance. Important research has been conducted to determine and adjust to optimal levels of latency, lag, jitter, and the like. Much less is said and understood about how to support interaction, communication, and collaboration. The goal of this research is to study the issue of collaborative awareness in immersive virtual environments like Tele-immersion. Collaborative awareness, in this context, is a concept that encompasses the caveats of one's knowledge about the CVE and its occupants. As a result of this study, software has been designed to provide support not only for collaborative awareness, but also for several other dimensions of collaboration.

1.1 The Connection with CSCW

Concepts are emerging in research on Tele-immersion which tie into the broader field of computer-supported cooperative work (CSCW) in new ways. The research is somewhat scattered however, and limited with respect to supporting collaborative functionality in CVEs. Even less research specifically addresses these issues with regard for Tele-immersion. Research on collaborative virtual environments, and on CSCW, more generally provides a framework for the current interest of collaboration in Tele-immersion.

1.2 Multidisciplinary Influences

Designing and building immersive CVEs requires the integration of many academic disciplines. As computing tools, collaborative CAVEs depend on high-speed computer networking, parallel and distributed computing, database management, graphics, software engineering, and human-computer interaction. But design of these systems also draws influence from psychology, human-factors, and other social and behavioral sciences. As the technology progresses, integration from mechanical and electrical engineering will likely occur (*e.g.*, for motion platform devices and wireless interface hardware). Tele-immersion has a wide ranging application domain and these applications will be influenced by many (seemingly unrelated) knowledge bases.

1.3 Scope of Research

The current work focuses on human-computer interaction and software engineering for computer-supported cooperative work (CSCW). The research approaches the design and implementation of Tele-immersive systems from a user perspective. There is interest in distinguishing between what supports and enhances interaction in Tele-immersion and what inhibits it. Many papers have been written that address the impact of overall system performance on usability of networked VR. Accordingly, attention to the wide range of human-factors issues is skewed in the direction of system performance. True user-perspective issues are sparsely represented in the research literature.

Many types of users are considered, with many different intentions of use. These include, but are not limited to:

- Scientific Visual Analysis and Design
- Military Simulation and Training
- Education and Distance Learning
- Architectural and Interior Design
- Entertainment

This list is just a sampling from the subject matter encountered both in the literature and from personal experiences at the Virginia Tech CAVE (VT-CAVE). But distinct from the unique, domain-specific requirements found in each of the categories above, are the common issues among them. One important commonality among all types of collaboration is collaborative awareness. Hence, there is a need to support user-awareness such that interaction among multiple users seems as effortless and "natural" as possible in the artificial environment.

1.4 Approach

An extensive survey of relevant literature in the areas of CSCW, collaborative virtual environments, and Tele-immersion provides the basis for making new claims and designing new applications. As components were developed, small pools of both experienced and inexperienced CAVE users were recruited to evaluate and test them. These users have included: experienced CAVE programmers, materials scientists with experience using the CAVE, architects and interior designers with only desktop CAD experience, undergraduate engineering students, and users with only moderate PC experience. Various others participated and tested prototypes in a less formal manner throughout the course of this research. These included HCI and usability specialists, but a great deal of casual information was collected from people touring the Virginia Tech CAVE facility.

Based on the findings from literature and the investigation into the practical needs of actual and potential users, several scenarios of use were developed. These scenarios are used to help illustrate how specific concepts might be realized in implementation. Bulleted summaries highlight dominant issues for each scenario and, in some cases, throughout them all.

The end result of this procedure is an application called the CAVE Collaborative Console, which continues to be developed. The CAVE Collaborative Console was developed at VT-CAVE as part of a team effort with Kent Swartz and John Kelso. The Console provides a direct interface to the CAVE for interacting with remotely connected users in a shared virtual environment. It is intended to be both the realization of the key concepts established by this and other research and a working model from which more complex tools can be built. Some of the features of the Console made their debut at Supercomputing '98. This experience provided one of the earliest assessments of the system. The Console has since been demonstrated for several government and industry organizations, including: NASA, John Deere, Science Applications International Corp., Applied Physics Laboratory, and GTE. Each has expressed a need for this type of technology in their various design, engineering, and simulation efforts. This exposure has also led to many practical insights concerning the type of work needing to be done and the tools desired to do this work.

Chapter Two:
Defining the Research: Background, Motivations, and Goals

With over 100 facilities worldwide now using and developing applications for Tele-immersion [URL 1], arguments of motivation are perhaps becoming moot. Indeed, this usage is not confined to academic research, but is gaining significant acceptance among a variety of industrial and governmental organizations. General Motors and Caterpillar were among the first to make use of this technology, and in the case of the former, even marketed a line of automobiles based on that experience. Since that time, John Deere, NASA, Searle, Motorola, Mercedes, and others have begun to include Tele-immersion in their design process (see URLs under the References section). The military is also beginning to investigate and include Tele-immersion to support some of its globally-distributed activities. For example, researchers for The Navy Collaborative Integrated Information Technology Initiative (NAVCIITI, pronounced "Nav-city") are investigating the use of Tele-immersion for collaboration between distributed battlefield command centers.

Nonetheless, Tele-immersion is not without its share of detractors. Given the substantial financial resources required of most immersive VR systems, this criticism is not entirely without merit. Base hardware for a first generation CAVE costs approximately $1.2M (USD), but administration and maintenance costs must also be added. Finally, we arrive at the cost of application development, for it is the application that must motivate the system. How, for example, can primary and secondary school systems hope to find funds for such systems? Yet primary and secondary education is often cited in the research to be ideal benefactors of Tele-immersion.

Quite often it has been observed that, after the initial excitement of experiencing immersion wears off, people start to wonder if the CAVE is not just a some kind of hyped-up *multimedia* system. Having logged hundreds of hours of use at the VT-CAVE for demonstrations and research, the VT-CAVE staff can attest to receiving this reaction on several occasions. HCI and usability specialists are quick to point out short comings in user-input devices and methods.

Despite whatever may be lacking, there is a clear and persistent effort being taken up to transform the way we interact with computer systems, and with each other *through* computer systems. To support this assertion, it is necessary to elaborate upon current and potential contributions of Tele-immersion to the types of collaboration this research is intended to support. To be certain, there are, and will continue to be, situations which do not warrant the resources required for true Tele-immersion. In some cases, Tele-immersion can be partially achieved through lesser systems. Regardless, this technology is being applied in practice and will continue to play a role in many types of computer-mediated, human-to-human interaction. It is perhaps for this reason alone that research should thus be motivated. Yet, the efforts required to support collaborative interaction in Tele-immersion has just started to scratch the surface.

One motivation of the current work is to reduce the disparity between aspiration and achievement; between what is desired and what is truly available. The research is influenced by the belief that human factors design should not be treated as if independent from an application's "core" objectives. This seems especially true for virtual reality, a paradigm dedicated to enhanced human-computer interaction. Add a collaborative component to the system and one can hardly disagree that human factors issues are coincident with every Tele-immersive application's objectives.

2.1 Domain-independent Benefits of Tele-immersion for Collaboration

Leigh, et. al. cite the ability to provide a "mechanism to support long-term collaboration" as a problem facing the entire VR community [1998, p. 1]. They point to "computationally demanding and data-intensive topics" such as: "global climate change, protein structures, nano-materials, video streaming, and interactive virtual reality" as example application domains that could benefit from Tele-immersion [*ibid*., p. 1]. Indeed, Tele-immersion has some general advantages which will be useful for each of these application domains:

> <u>Geographically Distributed Team</u>
>
> Collaborators who are separated by great geographic distances can work as if co-located [Finholt and Olson, 1997]. High speed networks are in place to support global collaboration with increasingly lower levels of latency and increasingly higher levels of fidelity (STAR-TAP, i-Grid) [Leigh, et. al., 1998].
>
> <u>Geographically Distributed Resources</u>
>
> Remote resources continue to play a greater role in computing as a whole. Increased access to one-of-a-kind facilities will depend on high-bandwidth and powerful supercomputing [Leigh, et. al., 1998, Smarr, 1999]. Each of these requirements is supported by Tele-immersion.
>
> <u>Supporting Large, Multi-dimensional Data Sets</u>
>
> CAVE technology provides a unique and necessary means of visualizing large and complex data systems that users cannot find in other systems. [Kriz, et. al, 1999] Complex geometric data is rendered more naturally in three dimensions.

High Levels of Presence and Fidelity

The presence of remote collaborators is made easily recognizable through the use of avatars [Bowers, et. al., 1994]. A strong sense of environmental presence is created by high fidelity, immersive models. Users can have opportunities to experience environments that might be otherwise inaccessible [Roussos, et. al., 1998, p. 1 of 17].

Unique Support for Views

Personal views convey spatial and organizational information in a natural manner. Tele-immersion supports multiple subjective views, which allows many collaborators to have their own, unique view of the information [Snowdon and Jää-Aro, 1997]. Personalized, single user views can also be also *shared* with this technology.

Gesturing

Tele-immersive collaboration supports a useful range of human gesture, which can reduce a dependence upon artificial turn-taking protocols [Bowers, et. al., 1996, p. 389].

Physical Attributes

Compared to head-mounted displays, CAVEs support the user in perceiving his or her own body in the most natural way (i.e., the user can see his or her own body while immersed). Stereoscopic shutter glasses are less intrusive and weigh much less than HMD's [Roussos, et. al., 1997, p. 2 of 17].

Still, it can be useful to identify some specific application domains for which Tele-immersion can, and is, playing a key role.

2.1.1 *Tele-immersion for Scientific Analysis and Engineering*

Some of the most compelling remarks on the need for technologies like Tele-immersion came on January 13, 1998, from NASA Administrator Daniel S. Goldin, in a speech entitled "Tools for the Future":

> The Department of Defense has a program that is the starting point for how we link diverse teams together in a simulation-based conceptual design environment. But we can take it a step further...into a high fidelity...high information content...distributed virtual environment. We can have a team in the northeast...a team in the south...and a team in the west...all working together on the same project in a virtual design space. Instead of taking the "Red Eye", teams can come and go electronically. More importantly...this provides us with something that has

been missing for too long. Scientists and engineers can work together as part of a collaborative team in the engineering design process. And they can do so while staying in their own offices and laboratories.

Figure 1: Collaborative Engineering Environment - NASA scientists working in front of an I-Wall [Source: ISE homepage, http://ise.larc.nasa.gov]

Even before Goldin shared this vision, Finholt and Olson had used the term *collaboratory* to refer to a "computer-supported system that allows scientists to work with each other, facilities, and databases without regard to geographic location" [Finholt and Olson, 1997, p. 28]. The objectives of the collaboratory outline a situation for which Tele-immersion was intended. Considering also the increasing resource requirements of modern scientific visual analysis and engineering, it seems reasonable to suggest that Tele-immersion can assume a positive role in enabling computer-mediated collaboration in the modern era.

In Europe, engineers in the automotive industry are using immersive VEs for tasks ranging from "immersive design review" to "immersive post processing" [Scharm and Breining, 1999]. Immersive virtual environments are also cited for use in the design review process of building power plants [Ebbesmeyer, et. al., 1999]. But requirements and methods for supporting this collaboration are still very much to be decided. For example, Ebbesmeyer, et. al., are quick to point out the multi-user nature of the design review process [p. 148], but fail to mention anything about multi-user support in the list of 10 system requirements [pp. 149-150].

2.1.2 *Tele-immersion for Systems Simulation*

The military is actively researching ways to deploy Tele-immersion for mission rehearsal, tactical management, battlefield visualization, simulation, and training. The Naval Research Laboratory has investigated the use of several types of immersive VR systems to support collaborative battlefield visualization [URL 2]. Searle, a pharmaceutical research company, is developing and testing a "Virtual Reality Tele-collaborative Integrated Manufacturing Environment" (VRTIME) "targeted at industrial process design, operation, and control." Among the more important goals of this type of research are better analysis and decision making techniques [URL 3].

2.1.3 *Tele-immersion for Architectural and Interior Design*

At Virginia Tech, the Interior Design Futures Lab (IDFL) holds an active interest in the CAVE. The IDFL is one of several remote CAVE labs situated across the campus. (The remote labs use I-Walls and desktop simulators to connect to the CAVE.) The goal in Interior Design is to illustrate full-scale, three-dimensional interiors concepts and to dynamically reconfigure them in a collaborative design effort. Immersive VR models are also useful for architectural "walk-throughs" and design review meetings. The potential user population consists of both design professionals and their clients.

2.1.4 *Tele-immersion for Education and Training*

Education and training are often cited in the literature as excellent domains for Tele-immersion. This has been demonstrated on several occasions. Through the NICE project, Roussos, et. al., have illustrated some benefits that Tele-immersion provides to elementary school students [1997a, 1997b, 1997c]. Formative evaluations of learning in immersive environments have revealed higher retention of abstract information [Roussos, et. al., 1997c, Carlucci, 1999, Iskowitz, 1999].

These findings are likely to prove useful in all kinds of complex learning situations, and for all ages. Students and teachers are able to interact in new ways. These will include sharing the trainer's own ego-centric view of a simulation, following the trainer's actions as he guides you through a simulation, and interacting with the trainer as if he were part of a simulation. When the trainee has trouble understanding a concept or acquiring a skill, the trainer will be empowered with these new methods for instruction. Training exercises which are costly and/or dangerous can be conducted in VR with high levels of realism [Sagar, et. al., 1994]. Within the integration of haptic feedback devices, interaction with the model will trigger realistic behavior. Finally students have opportunities to "visit" places that are no longer existent or are otherwise inaccessible [Roussos, et. al., 1999, URL 4, 1998, p. 1 of 17].

Figure 2: An Avatar in the NICE Garden [Roussos, et. al., 1997]

2.1.5 *Beyond Desktop Groupware*

Some of what has been learned about "traditional" desktop groupware can be applied to Tele-immersion. In fact, some of the issues governing collaboration in immersive environments are mere extensions of what has been learned in CSCW. On the other hand, Tele-immersion is quite different than any flat-display, desktop application and it is beginning to offer many distinct advantages over desktop groupware. This is not just in the form of high-tech meeting forums. Design is taking place and unique analytical techniques are emerging similar to the examples listed above. Tele-immersion represents a true paradigm shift in the way we view data, create models, conduct simulations, and interact with the computer and with one another.

To elaborate on this assertion, the following two sections respectively cover a configuration description for some collaborative IPTs, as well as some scenarios which highlight support and design considerations for Tele-immersion.

2.2 Configuration of the Collaborative Session

For situations which require "true" Tele-immersion, multiple collaborators share virtual environments from networked CAVEs. But systems need not be restricted to CAVEs alone in order to retain some qualities found in immersion. In fact, there a number of ways to configure smaller projection-based systems and even desktop workstations to facilitate collaboration in three-dimensional, semi-immersive VEs. To some extent these smaller systems can even provide the sense of "being there." Smaller systems may also be better suited for switching between tasks which require some immersion and tasks which require absolutely *no* immersion (e.g., document editing). The most common configurations are among CAVEs, Immersa-Desks (I-Desks), and desktop CAVE simulators.

2.2.1 *The CAVE*

The original CAVE design [Pape, et. al., 1997] is a 10'X10'X10' "room" with 3 walls (front, left, right) and a floor. The walls are made by stretching a single piece of mylar around a steel box frame. The floor need only be a flat, white, non-reflective surface. Stereoscopic images are rear-projected onto large mirrors, which re-orient the light through the translucent mylar screens. There are a total of four projectors which provide images for each of the four viewing surfaces: 1 projector behind each of the 3 walls, and 1 projector, which hangs from the ceiling, for the floor. The double-image which the projectors emit is made to appear voluminous through the use of *shutter glasses*. Shutter glasses block the passage of light, alternately between each eye. Stereoscopic emitters, placed in several locations around the frame, synchronize the shutter glasses with the update rate for the screens (120Hz or 96Hz).

The user's motion is detected by two tracking devices, each with six-degrees-of-freedom (DOF). These trackers are controlled by a magnetic system called the "Flock of Birds". One tracker is attached to the shutter glasses and monitors head motion. The other tracker is attached to the hand-held input device and monitors the motion of the hand which holds the device. Trackers communicate with the CAVE application to determine the correct direction and orientation for rendering the environment. These trackers, when attributed to avatars, can also facilitate the communication of non-verbal gestures, such as head-nodding and hand-waving.

The main input device for the CAVE is the *wand*. The wand is similar to a mouse in that it has three buttons and a simple steering mechanism. Often, however, the keyboard must also be used to provide input to the CAVE.[1] This requires the user to physically leave the immersive environment and return to the terminal. For certain applications this setup is sufficient. This may be especially true for the I-Desk, where the user can sit in front of the projection screen while remaining within arms reach of the terminal. But as Tele-immersive technology matures, the inadequacy of these input devices will become more apparent. Already there is a significant *gulf of execution* [Norman in Carroll, 1991, pp. 23] between the user and the application which can be highly attributed to a lack of proper input channels. This input problem becomes a key requirements when one is attempting to build collaboration into the system (see "Voice-Recognition Input" in Chapter 4).

[1] Other input devices exist (data gloves, joy sticks), but successful use has been limited.

The "Gulf of Execution" in the CAVE

"The gulf of execution refers to the difficulty of action upon the environment..." Donald Norman

Designing interaction into applications is arguably the single greatest concern for Tele-immersion. This is true for almost all CAVE applications, but is quite a hurdle when one adds the burden of human-to-human interaction to a system that barely supports human-to-computer interaction.

The primary means for interaction is the wand. Modeled after the mouse, the wand is essentially a point-and-click and grab-drag-and-drop device. As an input device, the wand can really only be useful as a means for interacting with some kind of graphical interface, equipped with buttons, menus, sliders and other GUI widgets. (These GUIs are not pervasive in the CAVERNUS community, however.) In most applications, there is but a mapping between a single button and a single action. Of course, this makes consistency among applications a near impossible achievement. If the keyboard can be brought into the CAVE (*i.e.*, a wireless keyboard), it might be considered another useful input device because of its familiarity. Finally, some recent and promising developments include wireless personal data assistants (PDAs) [Watsen, et. al., 1999] and hand-held "palettes" [Williams, et. al., 1999] as input devices.

While wand and keyboard methods may be familiar and arguably adequate for some tasks, we strive for interaction that has the same "realness" that Tele-immersion provides visually. Already, input devices exist that serve certain kinds of interaction well. Joysticks are well-suited to flight simulators, for example. Haptic, or force-feedback, devices reduce our reliance on artificial indications of intersection with objects that should behave as if solid.

In an informal survey of subscribers to the CAVERNUS listserv, success stories reporting these alternate input means were not common. In experiences at the VT-CAVE, we found ourselves dedicating a disproportionate amount of time trying to code for just the input methods. Little time was focused on the actual collaborative functionality that was to be invoked by said input. As a result of these circumstances, it seemed the best solution to the problem was to "ignore" it. In other words, the means of input was treated like a black box. The implementation described in Chapter 4 explains how to design for collaborative interaction in a way that is independent of the means of input.

2.2.2 *The I-Desk and CAVE Simulator*

The Immersa-Desk is essentially a scaled-down version of the CAVE. The I-Desk operates as a single surface, rear-projection system with a large, back-angled glass screen. Images are rendered stereoscopically just as they are in the CAVE. Rendering, motion, and interaction are also handled as in the CAVE; by shutter glasses, tracking sensors, and a wand.

The CAVE Simulator is a desktop version of the CAVE. The simulator runs on SGI Octane and O2 workstations and is rendered in an X window. With the addition of a stereoscopic emitter and shutter glasses, graphics can be rendered to appear voluminous, as if they were projecting in front of and into the monitor. By default, graphics are rendered monoscopically.[2] This is often sufficient for prototyping and testing; the most common uses for the simulator. The main difference between the simulator and both the I-Desk and the CAVE are the input and tracking devices. All input to the simulator is handled through a combination of keyboard short-cuts and mouse manipulation, in much the same way as is done in action games like Quake™ and Doom™. Tracking is handled internally by the application, based on the user's input from the mouse and keyboard. Though the user does not wear any tracking devices, gestures can still be simulated with the proper mouse and keyboard manipulations. This is an awkward process, but since the desktop configuration is familiar, this method seems sufficient.

All software written for the CAVE will run on both the I-Desk and CAVE Simulator without re-compiling or re-linking the executable. While not platform independent, this feature is powerful in two ways. First, the vast majority of programming time is spent working on an SGI workstation. Typically the CAVE is used toward the end of the design stage, to make refinements, and to run the finished program. Having an application that works with either hardware configuration makes this process quite easy. The second way this hardware flexibility is beneficial is that it allows people with different systems to work together in the same virtual environment with the exact same application. This is not to say that three users on three different systems will have the same experience. The immersive factor is one obvious difference. There are many considerations involved with building robust applications that are effective for all three hardware configurations. For example, objects placed at the same distance appear farther away on the simulator monitor. This is a subject not within the scope of this paper, however.

Other Immersive Projection Technologies

Since the release of the CAVE and I-Desk, several derivatives of these systems have been implemented. These include the 6-wall VR-Cube, which adds back wall and ceiling projection surfaces, the I-Wall (or Power-wall); a wall-sized version of the I-Desk, and the VisionDome; a spherical immersive system.

2.2.3 *CAVERNsoft and LIMBO*

CAVERNsoft

CAVERNsoft [Leigh, et. al. 1997] is the so-called "backbone" which supports Tele-immersion. CAVERNsoft provides a high-level programming interface that Leigh describes as:

> "a C++ hybrid-networking/database library optimized for the rapid construction of collaborative Virtual Reality applications. Using CAVERNsoft, VR developers can

[2] The CAVE and I-Desk can also run in monoscopic mode, but this is seldom done.

quickly share information between their applications with very little coding. CAVERNsoft takes over the responsibility of making sure data is distributed efficiently"[3]

Information about each remote user can be easily customized and propagated through the network, delivered either server-to-client, client-to-server, or client-to-client. More information about CAVERNsoft can be found at URL 5 in the references section at the end of this paper.

<u>LIMBO</u>

LIMBO [URL 6] is the layer between CAVERNsoft and the CAVE Collaborative Console, and is another integral part of this collaborative system. LIMBO provides the basics for enabling interaction over networks linked together by CAVERNsoft. Though LIMBO is described as little more than a "simple collaborative program that allows multiple participants, represented as avatars, to load and manipulate 3D models in a persistent virtual environment" [URL 6], it was the crucial starting point for the Console. LIMBO provides 3 main things: a generic 3D avatar with a moving head and hand, a pointer for selecting, grabbing and moving objects, and a generic world to occupy. There are also “hooks” for internet telephony. Beyond these features, it is up to applications developers to extend LIMBO into something more powerful. More information on LIMBO can be found at URL 6 in the references section at the end of this paper.

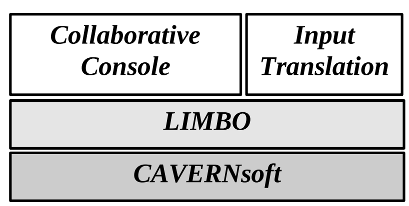


Figure 3: Layered Model of Software Modules

2.3 Scenarios of CAVE Use

It can be helpful to construct typical and/or potential scenarios which illustrate how Tele-immersion and other collaborative IPTs enhance our design and analysis capabilities. The following scenarios are grounded in work currently taking place in both research and industry but also describe what might be possible in the future.

2.3.1 *Remote Tactical Command Center*

The Navy is interested in using Tele-immersive collaboration technologies to view and orchestrate the battlespace from remote, distributed command centers. They hope initially to deploy such a system for

[3] http://www.evl.uic.edu/cavern/cavernsoft/intro/index.html

mission planning, mission rehearsal and course of action analysis, but desire to eventually create real-time environments which reflect real-time engagements and events. They envision a large-scale collaborative virtual environment (CVE), populated with ships, aircraft, weapons systems, armor vehicles, and even individual soldiers. Each object, or agent, maintains information about its state (e.g., geolocation, identification, payload, etc.) and can be coordinated by commanders from remote physical locations.

In this scenario, Commander One is immersed in a CAVE located ashore at Naval Air Station Norfolk. Commander Two is immersed in a CAVE on board U.S.S. Nimitz. The two are connected via a satellite-based digital voice and data network which enables them to co-inhabit a virtual theater of operations and communicate with each other in real time. For the current training exercise the CVE is populated by the Nimitz and its aircraft and escort ships, and a simulated enemy order of battle. Commander One views the environment as a *deity* [Leigh, et. al., 1996]. Thus, he sees the world from a bird's eye perspective. This puts him in a position to make high-level observations and decisions. Commander Two views the environment as a *mortal* and is in a position to make higher resolution observations and more fine-grained operational- and tactical-level decisions. Because the real-world environment encompasses thousands of square miles, both view their environment from a relatively large scale, but the scale can be changed, dynamically, to any zoom factor.

Commander Two is interested in the status of a flight (group of planes), call sign Broadsword, which is en route to a land-based target. Commander Two has *encapsulated awareness*; she can only see that part of the environment which can be tracked by her own local resources [Greenhalgh and Benford, 1997, p. 2 of 6]. She issues a verbal command to the system, "Computer, zoom sector seven and confirm!", and is immediately taken to sector 7 of the virtual battlespace. The system confirms the command by playing back, "Zoomed to sector seven." Commander Two continues, "computer, track Broadsword and confirm!" This extends a ray to the flight's avatar and displays a StatusPanel. The StatusPanel supplies Commander Two with important information regarding the flight; number and type of aircraft, total payload, geolocation and distance from target, etc. Having received an update from airborne early warning radar Eagle, Commander Two's CAVE shows 5 enemy aircraft approaching Broadsword. Commander Two verifies that Eagle has given Broadsword flight a "heads-up" on the approaching bogies.

Back on shore, Commander One is manipulating the simulation in real-time by making adjustments to the enemy's facilities. Commander One has global awareness, meaning he is always able to view any part of the environment from any perspective, through any available medium, at any time. Assuming the role of the Red Commander, Commander One controls all of the enemy's ordnance and equipment. Having launched the five aircraft now pursuing Broadsword, Commander One directs an Air Defense Battalion into a position to fire Surface to Air Missiles (SAMs) on Broadsword as it approaches. Meanwhile, Commander Two turns her attention momentarily to the target. She pulls up a StatusPanel and reads new intelligence information stored for that object and notes from communications and electronic intercepts that the enemy is repositioning SAMs to intercept Broadsword flight on ingress. She relays some of this information to Broadsword lead, and because of this new threat, orders EA-6B jammer aircraft and F/A-18 HARM anti-radiation shooters to protect Broadsword flight.

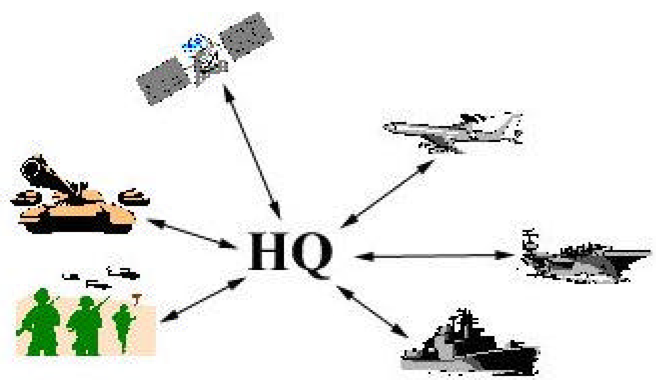


Figure 4: Littoral Battlespace for NAVCIITI

<u>Summary</u>

- ❖ This scenario captures several key elements of collaborative interaction: location awareness, global and encapsulated awareness, object manipulation, status information, and synchronous verbal exchange.
- ❖ This scenario describes rich interaction sequences which can be supported in Tele-immersion.
- ❖ This scenario was reviewed by a naval aviator with 26 years of experience, including as Airwing Operations Officer, Squadron Commanding Officer, Carrier Combat Direction Center Officer, Carrier Operations Officer, and Battle Group Operations Officer. The reviewer is also investigating the use of Tele-immersion in military simulation and collaboration.
- ❖ When not being used for this type of collaboration, each CAVE can be exploited for many other tasks (e.g., flight simulators, heavy machinery simulators).
- ❖ But, this scenario will be expensive and a challenge to implement. It relies heavily on wireless computer networks, absolute fidelity and credibility for sensitive data, and other formidable factors now being studied.

2.3.2 *Collaborative Engineering Environment*

The Collaborative Engineering Environment (CEE) is an immersive design and analysis concept being investigates by both NASA and the Navy. In the case of the former, the hardware to support the CEE is already being installed in NASA sites all over the country. As this phase of the CEE draws to completion, NASA is assembling teams of computer scientists and engineers to provide the software required to support this type of collaboration. NASA projects that they will be able to support scientists collaborating in immersive CVEs by the year 2004 [URL 7].

The general purpose of the Collaborative Engineering Environment is to allow geographically separated scientists and engineers to share a virtual environment in which they may conduct tests, make design decisions, and perform analyses. The advantages of immersive collaboration include the ability to manage large data sets, the increased fidelity with the physical world, and the added value of the third dimension.

In this scenario, three scientists, Drs. Smith, Jones, and Johnson, each from separate NASA sites around the country, share an immersive design environment. Today the scientists are discussing the latest prototype for a re-usable launch vehicle. The three scientists appear as if standing around a large pedestal in the center of a virtual laboratory. Dr. Smith has just completed several changes to the design and wishes to share them with his colleagues. He says, "computer load model one please," and a large model of the spacecraft appears before the group. The model is "tagged" in several places, indicating that changes were made to the model. Each tag stores the owner of the change and the date and time of when the model was last changed.

Each user has a unique personal view of the model relative to where that person is "standing" around the pedestal. As the group co-manipulates the model, each person sees these changes (e.g. rotation/orientation) according to their view. Each person can also view the model from several distinct perspectives using the `camera` feature. Dr. Jones wants to get a look at the model from the right side. He issues the command, "computer view model from right side please," and immediately switches his view to the be facing the model's right side. This does not affect the views of Drs. Smith or Johnson.

Desiring to document his presentation for future reference, Dr. Smith issues another command. "Computer start VCR record please." Back at the terminal, a video cassette recorder begins recording the session from the view of Dr. Smith's shutter glasses. Now Dr. Smith wants to steer his colleagues through the model while pointing out his changes. He issues the `tether` command which will attempt to "attach" the other two participants to his position. First Dr. Smith must have permission to seize navigational control from the other two. Drs. Jones and Johnson each receive tether requests from Dr. Smith, which they must, and do, confirm. Drs. Jones and Johnson are then "tethered" to Dr. Smith, who leads the group through the model.

Summary

- ❖ This scenario captures several key elements of Tele-immersion: accessing stored files while immersed, archiving the collaborative session, shared object manipulation and design, and synchronous verbal exchange.
- ❖ This scenario also illustrates the concept of "tethering", which allows users to steer one another through an environment. Commands like "tether", which are sent from one collaborator to another, require confirmation before they are executed.
- ❖ The "tags" illustrate a form of awareness of environmental change.
- ❖ This scenario is typical of what will be expected for the NASA Intelligent Synthesis Environment initiative, which includes the CEE as a major component.
- ❖ This scenario still requires the implementation of a seamless, immersive interface between the computer and the VCR. Currently users can launch media recorders on the computer monitor, but these digital movies consume large amounts of space.
- ❖ This scenario is not ready for finely-detailed, mission critical design and simulation. NASA's earliest projection for when their engineers will work together in immersive CVEs is 2004.

2.3.3 *Training to Perform a Skilled or Cooperative Task*

Sagar, et. al., demonstrated how extremely high-fidelity VR can be used for training surgeons to perform delicate procedures on the human eye [1994]. Airlines and the military have long used flight simulators of various complexity to train pilots. Many also believe that Tele-immersion has potential to contribute positively toward training for cooperative tasks in three-space. Tele-immersion may also enhance student-instructor interaction, allowing the instructor to monitor or test the student in new ways [Schiefele, et. al., 1999, Roussos, et. al., 1997].

In this scenario, Federal Disaster Relief specialists are training for bomb scare disaster exercise. The situation involves a team of three trainees whose task is to locate and secure a biological weapon in a suburban mall. The team members physically distributed among three I-Desks, but are co-located in the virtual environment. At times, the team members are very near one another and at other times they are far apart in the virtual space. They use internet telephone to simulate radio communications among them. The trainees also have simulated access other devices they might normally use for the task, but they have no other artificial tools to help them. The trainees interact with their devices and their environment through data gloves, which are both tracked by the system.

There is also a trainer in this environment whose presence is only sometimes known to the trainees. At random opportunities, the trainer can either assume an avatar designating him to be the trainer, remain invisible while only observing, or assume the role of some other actor in the environment (*e.g.*, a victim).

The trainer also has access to various radar and status information for tracking and monitoring his trainees.

Throughout the course of the training exercise, the trainees navigate through the environment while talking to one another over their simulated radios. As the trainees move through the environment they have to open doors, climb stairs, etc., just as in a real environment. Occasionally, a trainee will interact with other avatars, which may or may not represent actual people. The avatars may give information to the trainee or perhaps even hinder the trainee's progress toward his goal.

Summary

- This scenario shows how instructors can interact with their students in new ways. The trainer can switch easily between modes of passive observer, participant, or active instructor.
- This scenario shows how radar can support awareness of location when the observed moves out of sight of the observer.
- It is not clear, based on this scenario, how participants navigate through the environment using the data gloves.
- Investigation on behalf of the Federal Emergency Management Association into Tele-immersion has only just begun. Its true objectives have not been defined (or at least made public) with regard to immersive collaborative training. This scenario can only illustrate highly abstract and generic tasks.

Chapter Three: Collaborative Awareness in Immersive Environments

The concept of collaborative awareness may be the most fundamental aspect of Tele-immersive interaction. Without at least some minimal indication of the existence of other collaborators within the environment, it is extremely difficult to establish even a foundation for any kind of interaction. Further, it is not just important to be aware of others, it can be important to know their awareness of you. In this way, we are cognoscente of our obligation to others but also can maintain our privacy [Isaacs, et. al., 1996, p. 319]. Finally, a person's level of awareness about the environment can have a direct effect on the collaboration [Ellis, et. al., 1991, p. 55]. When working asynchronously, for example, collaborators should be made aware of any changes which may have occurred since the last time they were in a particular environment. In fact, awareness is viewed here as a set of issues that concern one's knowledge about the states of other collaborators within a virtual environment and that of the VE itself.

Collaborative awareness has long been an issue for CSCW. In their 1991 paper "Groupware: Some Issues and Experiences", Ellis et. al. alluded to many elements which might now be placed under a heading of "awareness." These include: "group focus and distraction issues" [p. 50], and "notification" [p. 55]. There are numerous other papers, under various auspices, which address some aspect of collaborative awareness. Seeking to increase the capabilities and usability of collaboration-transparency systems in Java™, Begole turns to CSCW to provide guidelines [1998, pp. 3, 8, 10-21]. Most of the items he discusses under the heading of "Groupware Principles" deal with collaborative awareness.

The work of Benford, Bowers, Greenhalgh, and others [Benford, 1994, Bowers, 1996, Greenhalgh, 1995, 1997] has considered the impact of awareness for the specific case of collaborative virtual environments. Their work has provided a great deal of support for the transition into Tele-immersion and other IPTs. To help readers realize this contribution, it summarized below. Following the summary, extensions to their model are introduced and explained.

3.1 The Spatial Model of Interaction

Awareness can be achieved by numerous means and through various media, each of which may have individual or combined implications. Awareness can be either mutual or one-way. It can have a wide or narrow scope of interest and extended or limited range. A participant's desire to be noticed can be enhanced or restricted, thus increasing or decreasing the likelihood that other's will be aware of that participant. One model that accurately captures these nuances for VR is the Spatial Model of Interaction [Benford, et. al. 1994, Greenhalgh and Benford, 1997].

The Spatial Model represents interaction based on the physical properties of space. Thus, the abilities to see and to hear are affected by distance, direction, and possible obstruction. The main concepts of the Spatial Model are:

<u>Medium</u>	the mode of communication; text, audio, video, etc.
<u>Aura</u>	the portion of space for which interaction is enabled and allowed
<u>Focus</u>	a representation of the "observing object's interest in a particular medium"; what the observer is trying to perceive relative to the observer's *range* of perception
<u>Nimbus</u>	a representation of the observed object's desire to be perceived in a particular medium; *i.e.*, the range at which a person can broadcast his or her own presence so that others may direct their focus toward that broadcast
<u>Adapters</u>	objects which can modify aura, focus, and nimbus in order to customize interaction between participants

Essentially, aura can be thought of as a boundary containing one's *range of perception* (focus) and *range of projection* (nimbus). Adapters can then be used to increase the range of either component (*e.g.*, a virtual microphone increases one's verbal range of projection, virtual binoculars increase one's visual range of perception).

Some Additional Terminology

<u>*Transportation* and *Fidelity*</u> **-**

Transportation is the degree to which the user feels he has moved beyond his local environment into a remote, shared environment. Fidelity, also referred to as *artificiality* [Benford, et. al., 1996, p. 79], is the degree to which the virtual environment either mimics the real world or appears synthetic [*ibid.*]. Both cater to the user's perception of "realness" in the VE.

<u>*Spatiality*</u> **-**

Is "the degree to which [VEs] support key spatial properties of containment, topology, navigation, and a shared frame of reference" [*ibid.*].

The Spatial Model seems a likely starting point for extending this discussion of awareness into Tele-immersion. As a result of local experiences with the CAVE, it is necessary to explain the assumptions, modifications, and extensions made in this paper with regard to the Spatial Model.

3.2 Extending the Spatial Model

All objects in a virtual environment have their own aura, focus, and nimbus. According to Benford, et. al., interaction between two objects is not defined until their auras intersect. The term “interaction” is taken to mean any sort connection or communication between objects. Thus, interaction may be as simple as the mere sight of another object, or it may be as complex as a face-to-face conversation. Once intersection of two or more auras has occurred, each object’s own focus and nimbus determine, respectively, the degree to which they *can be* aware of other objects and the degree to which other objects *can be* aware of them. This *potential* for awareness is emphasized because it is possible for someone with a large focus to be unaware of someone with no nimbus. Likewise, someone projecting a large nimbus may go unnoticed by someone with no focus toward that nimbus. In accordance with what we expect in the physical world, aura, focus, and nimbus are all defined based on the nature of the medium. For example, most people can see much farther than they can hear, and so the *default* aura for a visual medium is expected to be larger than the *default* aura for an audible medium. Again, the word “default” is used to remind us that we are dealing with a computer-mediated environment and that it may be desirable to artificially manipulate some particular component of our awareness. For example, there need be no restriction on focus that requires it to be in the same location as one’s presence. This makes it possible to share views or move one’s view to arbitrary locations in space. Similarly, privacy can be maintained by combining a small or non-existent nimbus with a limited focus.

Gradient Awareness

Awareness is a scaleable concept comprised of multiple components in multiple media. Awareness, particularly aura, focus, and nimbus, can be manipulated by auxiliary interface devices. Put simply, if a participant is able to perceive all possible awareness cues in an environment, from all possible sources, then that participant is said to be *totally aware*. If a participant does not or cannot perceive any awareness cues, despite the transmission of such cues, then that participant is said to be *totally unaware*. The Spatial Model certainly allows for gradient awareness in the sense that focus and nimbus can be dynamically redefined by the user. But each medium may have multiple parameters which contribute to one’s level of awareness in that medium. Thus, it is possible for someone to be totally aware for some media and totally unaware for others. Likewise, it is possible for someone to be *partially aware* in all media.

Within the context of total and partial awareness is the user’s level of *global awareness*. Global awareness means that the user is able to perceive any part of the world, from any perspective, at anytime. The globally aware participant has all available communications media at his or her disposal. A user who has *encapsulated (or situational) awareness* is restricted to awareness of particular area in the environment. From a system point of view, control over the user’s of awareness is sometimes referred to as *interest management* [Abrams, et. al., 1998]. Again, much of what is known about interest management speaks to issues of performance.

The current work has analyzed the overall concept of awareness into 6 main categories. These are as follows: *Action Awareness*, *Attention Awareness*, *Environmental Awareness*, *Location Awareness*, *Awareness of Presence*, and *Views*.

3.2.1 *Awareness of Presence*

The term "presence" has often been invoked under a broad definition. Generally, presence is the feeling of "being there." In the context of the environment, presence deals with the degree to which you feel a part of some virtual space; that the space exists and you are occupying it. The presence of remote collaborators or objects in the virtual space is mostly and issue of embodiment or some peripheral indicator (participant list, radar, etc.). This kind of presence provides the sense that other collaborators, and even other objects, share the environment with you. One must know that other collaborators exist before they can be included in any form of interaction. Basic indications of presence are not difficult to achieve. A simple list of each participant's name is enough to tell you that others share your environment. Likewise, a textual description of the world around you is an indication of presence (*e.g.*, Multi-user Dungeons, or MUDs). These methods are computationally inexpensive and satisfy the basic need. But textual information does little more to qualitatively reaffirm the presence of space and others who may being sharing that space with you.

In Tele-immersion, which relies on a supercomputing and very-high bandwidth architecture, presence can rise to a new level. Tele-immersion places the user "inside" the environment, and the way the user views the environment is much different than other kinds of CVEs. *Avatars*, graphical representations of remote collaborators, are examples of embodiments which establish a strong sense of presence. Avatars are not unique to Tele-immersion and, even in the CAVE, can be as simple as GIF images of the other participants. On the other hand they may be as detailed as live, three-dimensional video stream images of an actual person. While such an avatar is not found in common use, powerful avatars do exist which lend themselves well to gestural communication. Accordingly, collaborators in an immersive VE can be represented by embodiments that exhibit features rarely found in traditional groupware.

Through various levels of detail, particular attributes of the avatar can be used to stimulate different notions of presence and even other types of awareness. In Tele-immersion, we can simply look at another avatar and instantly determine many things that otherwise might require complex and artificial indicators in more traditional groupware. If a remote user is distant from us in the virtual world, then her avatar, obeying the natural laws of space and light, appears small and difficult to see clearly. As the user moves closer, or we move closer to her, the avatar looks correspondingly larger and other details of appearance become clearer. When the avatar is standing directly in front of us, we see a life-size embodiment almost as if the person were there with us.[4] These same things are also true for the presence associated with the environment. Standing at the base of a virtual 40-story building has quite a different effect than reading a description of the same scene, while perhaps viewing an image of the building.

Sound

Sound can also be an important component in perceptions of presence. In face-to-face interaction, we communicate by speaking. What we hear is strongly coupled to what we see; the hands waving, the lips moving, the eyes glancing around the room. All of these things together contribute to a greater feeling of presence. In Tele-immersion, the third dimension allows us to exploit *multi-directional* sound even further. Already, much has been done to combine multi-channel sound with volumetric, virtual space so that users correctly perceive sounds based on distance, direction, and amplitude [URL 8].

Such advances are not without limitations, however. A live, three-dimensional video stream, for example, may require more bandwidth than is currently available for practical uses. More generally, there should be some relationship between the complexity of the avatar and the task at hand, so that a minimum set of elements are used to effectively convey presence [Leigh, et. al., 1997].

Still, as avatars more closely emulate humans, they have the potential to support other forms of awareness(*e.g.*, action and attention). Through the use of motion tracking devices, avatars have been used to convey simple gestures like eye-gaze and hand-waving [Benford, et. al, 1994, Leigh, et. al., 1997]. Benford's, *et. al*., first implementation of MASSIVE showed how visible changes to an avatars appearance provided cues that related the remote users ability to communicate via two different media. When one remote user was able to hear what other remote users were saying[5], his head-shaped avatar displayed ears. When the remote user was able to communicate verbally, his avatar displayed a mouth [Benford, et. al., 1994]. These are some examples of attention awareness.

3.2.2 *Attention Awareness*

Attention awareness deals with a person's focus and range of perception. In simple collaborative exchanges like on-line chat or UNIX talk, it is practically impossible to determine a remote collaborator's focus of attention. The impact of low attention awareness on the collaborative session was illustrated in a study of MASSIVE [Bowers, et. al., 1996]. Through Conversation Analysis, Bowers, et. al. showed how

[4] Not all avatars are life-sized. Again, this is dependent upon the nature of the application.
[5] Through peripheral hardware (speakers, headphones, microphones, cordless telephones).

a lack of perceptible cues caused confusion in the session when one member diverted his attention to a colleague in the real world.

Providing perceptibility of another person's focus of attention seems to one of the more desired, yet allusive, goals of collaborative VR. Attention awareness can be increased through the use of avatars, which are animated by motion tracking sensors. Attention might be even more acutely perceived if eye tracking sensors are also used, but studies are certainly needed to determine what is necessary and enriching.

Awareness and Synchronization

The need to synchronize a user's virtual embodiment with his or her actual focus, or even participation, was documented in Bowers', et. al. study of MASSIVE [1996]. In their report, they describe at least two scenarios which demonstrate this point. In the first incident, collaborators in the virtual world found it difficult to determine when their colleagues were turning attention toward collaborators in the real world. When a person shifted his attention to a real-world distraction, no indication was given in the virtual world that this event had occurred [pp. 383-385]. Eventually an informal and *artificial* method was adopted which required the user switching focus to position his or her avatar to appear in a somewhat prone position. The second incident was referred to as "corpsing" [p. 383]. In this case, a remote collaborator's network connection would go down but his avatar would remain as a "corpse" in the virtual world. Those remaining in the virtual world were often found trying to interact with the "ghost" avatar.

The problems described above have been observed in the LIMBO/CAVERNsoft implementation and have required additional consideration in the design of the Collaborative Console. The "ghost" avatar scenario must be accounted for, not only to remove the avatar from the environment, but also to remove any other indications of that avatar's presence (*e.g.*, radar blip, name on a list).

Switch Management - Exchanging Focus Between Real and Virtual Worlds

Bowers, *et. al,* cite the need for protocols which handle issues arising from collaborators "switching" focus between the virtual collaboration and something requiring attention in the real world [1996, p. 384]. When designing for scenarios which involve switching, Bowers, et. al., propose some form of "switch management" [p. 384]. Switch management can be a subtly complex issue. An example above illustrates how simple switches can easily occur which are totally unnoticeable in the virtual world. So, it would seem that switch management is not so much concerned with how the local user executes a change in focus, but rather *how remote users are made aware of this change*. By adding this dimension of awareness, however, we create a situation which still requires some input in order to be affected. In the experiences reported by Bowers, et. al., this input came from the user, who made a verbal announcement and repositioned his avatar to appear as if lying down.

Bowers, et. al., warn against methods that might interfere with the session [1996, p. 384]. On the other hand, it could be argued a switch method should be no more nor less intrusive than any real world interruption of a collaborative session. Personal interactions between two or more people are often interrupted in reality. A meeting might be interrupted by a phone call or a knock on the door, for example. Interruption, by its very nature, requires some sort of "awkward" intervention. This could be a simple announcement to other collaborators that you are changing your focus to something in the real world (e.g. saying, "Excuse me for a moment while a take this call."). But there also seem to be occasions when a change in focus of attention should also be accompanied by some corresponding change in presence. In real life, one may asked to be excused from a room, *and then physically leave the room*. In a virtual environment, one might "gray-out" his avatar while attending to other business.

3.2.3 *Action Awareness*

Action Awareness concerns the ability to know what other collaborators are *doing*. To some extent, different looking avatars can be used to convey different actions of remote collaborators. Avatars can also be animated with motion feedback from tracking devices. In this way, it is not difficult to show when someone's head is turned or in which direction they might be pointing.

Physical Gestures

Physical gestures play an important role in actual, face-to-face interaction. Shrugging one's shoulders, waving one's hands about while talking, and the many other elements in the vocabulary of human body language are indispensable in this type of interaction. Likewise, there is reason to believe that gesturing can be especially useful in Tele-immersion. It has been demonstrated that head position and orientation, body direction, and hand position and orientation are enough to convey important physical gestures [Leigh, et. al., 1997]. It seems simple, yet profound that we can share artificial environments, separated by vast physical distances, and still be able to point to a shared model and say, "look over here", indicating some common frame of reference for all users.

Other Changes in an Avatar's Appearance

As described earlier, MASSIVE demonstrated that simple changes in an avatar's appearance provided a sufficient means for viewers to determine which communications channels are available to a remote user. Leigh, et. al. further demonstrated that other changes could be made to an avatar to convey a remote user's actions. They created two different concepts, "mortal" and "deity", which showed that even basic avatars could support action awareness through embodiment. The primary motivation for using the two modes is so that one can make gross or minute manipulations depending on whether one is acting as a deity or mortal, respectively. Depending on whether a participant needs to make gross or minute manipulations to the environment, he or she will switch between modes. The mortal mode presents the environment to the user in fine detail, as if he or she is in the world and a part of it. This allows the user to manipulate small objects. The deity mode presents the environment in gross detail, as if the user is looking down on the world. This allows the user to manipulate large objects, and even the entire environment.

A change in mode is reflected by a change in the size of the avatar that other collaborators perceive. Mortals appear to be "life-size", while deities loom overhead and are clearly much larger-than-life in size. This drastic change in presence indicates to other collaborators a change in action from minute manipulation to gross manipulation. By changing the size of the avatar according to the mode of manipulation, Leigh, et. al., demonstrate support for action awareness.

3.2.4 *Environmental Awareness*

Distinct from environmental *presence*, this kind of awareness deals with the user's ability to perceive *changes* in the environment. We are especially interested in changes that are the result of some other participant's actions in the virtual world. However, environmental awareness is not the same as *action* awareness. Because environmental changes can take place over the course of several asynchronous sessions, environmental awareness is not of another's actions. It is the awareness of the *results* of a collaborator's (or the system's) actions and how they may have modified the environment.

Some methods for supporting environmental awareness between asynchronous sessions were discussed at the 1999 Immersive Projection Technologies Workshop (IPTW '99). A representative from General Motors raised the issue of version control over shared models. He was interested in some means of explicitly denoting asynchronous changes to a model so that everyone sharing the model would be aware of these changes. The GM representative went on to explain that, in the desktop CAD (Computer-Aided Design) environment, engineers are already familiar with cues like "virtual editor's marks" and "virtual post-it notes". Citing Leigh, this author agreed that similar "tags" might be useful in Tele-immersion. In this way, one engineer might make a voice recording or "take a snapshot" and then place a hyperlink in the environment so that others could access this information [Leigh, *et. al*, 1997, p 3 of 23].

It is also possible for environments to change on their own (see "Persistent Tele-immersion", below). For example, time could change from day to night or *vice versa*. Other changes, however may be less obvious. In an extremely large VE changes in one part of the world may go unnoticed in other parts. If a person leaves the virtual environment, but the environment persists and changes, to what degree will the user be aware of these changes? Items such as these are topics to consider when deciding how to support environmental awareness.

3.2.5 *Location Awareness*

Location awareness is knowledge of both your position in the virtual world and the position of others in that world. Location awareness is one of the more easily implemented attributes of collaborative interaction. A person's location in a virtual environment is nothing more than three floating point numbers representing their (x, y, z) coordinate position. If desired, orientation can be added for the cost of three more floating point numbers. This information can be used to provide simple locators, such as radar, for finding collaborators or objects.

Other heads-up displays can increase our knowledge about the location of remote participants. Textual information, in the form of participant lists and status panels might provide, for example, the physical location of the remote, where the user is in the model's time[6], or what time it actually is at the remote user's physical location.

An Experience With Location Awareness

At Supercomputing '98, this author was fortunate enough to participate in a global Tele-immersion demonstration [Leigh, Johnson, 1998]. During this demonstration, I shared a large, multi-room CVE with as many as 10 remote collaborators physically located at different points around the globe. Each "room" of this virtual "atrium", as it was called, showcased a different virtual environment.

The lack of, and consequent need for location awareness was quickly apparent. Most of the atrium's rooms were also large-scale, multi-room models. Only the contribution from the VT-CAVE included support for location awareness. Upon entering an immense model of a Silk Road shrine, each at different times, several of us were left blindly wandering around in search of and calling out to one another. To overcome this frustrating problem, we were forced to exit back into the atrium, agreeing to re-enter the shrine in quick succession. Fortunately, we could exit quickly with the press of a button. In essence it was like resorting to the classic "reset" to resolve a problem.

Upon entering the VT-CAVE model, on the other hand, users were greeted by heads-up displays of both a participant list and a radar. The participant list made it easy to tell who was in this part of the environment and how "far away" they were. At any time we were not able to see a remote user's avatar, we could easily locate that person with the help of the radar.

[6] Some applications, such as CAVE5D [Wheless and Lascara], allow users to move forward and backward in the model's timeline while otherwise working concurrently.

3.2.6 *Views*

A person's literal point of view has been a driving source of inspiration for research in Tele-immersion. As a visualization medium, Tele-immersion provides unique frames of reference for viewing graphical data. As a collaborative medium, Tele-immersion can further leverage this advantage. The need to support individual and "subjective" (personal) views for each participant has been recognized [Begole, 1998, Snowdon and Jää-Aro, 1994]. Subjective views are, in fact, the natural mode for human-to-human interaction; everyone sees through his or her own eyes. Such is the case in Tele-immersion, but because Tele-immersion is a *hyper*-reality[7], means exist to support many different points of view.

Camera Positions and Locations

Just as any large building may be equipped with a surveillance system, a large-scale VE or model might include multiple *camera positions and locations*. Camera positions typically allow a user to view an object or location from different directions and orientations (e.g., top, front, back , left, right). Cameras can be placed at fixed locations in the environment, or can be specified dynamically by the user. Cameras are used in most 3D CAD interfaces and might similarly be useful in Tele-immersive design.

Shared Views

Shared viewing is literally seeing through another's eyes. In CSCW, shared viewing is a classic WISIWYS scenario ("What I See Is What You See"). One or more connected users sees the world in the exact direction, orientation, and manner as the person whose view they are sharing. An important issue for systems with view sharing is the tightness with which two or more views are coupled.

In some sense, the concepts of shared viewing and multiple camera locations begin to blur the line between awareness and navigation. WISIWYS guided touring and other methods of leading someone through an environment combine subjective and shared viewing with navigation and movement. Likewise, location awareness provides useful information for navigating a virtual environment. Tele-immersion, as an *hyper*-reality, provides further support for navigation and movement through methods which are not new to either hypermedia or CSCW. This idea is continued in the section 3.3.2.

3.3 Other Awareness Issues

Computer-mediated collaboration is a multi-faceted activity. Sometimes collaborative interaction is a synchronous exchange and sometimes it is asynchronous. Whether occupying a shared editor or a shared immersive environment, users also require some means for navigating the virtual space. Multiple participants sometimes have specified roles and relationships (*e.g.*, team leaders and their teams,

---

[7] Credit to Dr. Richard Nance, of the Virginia Tech Department of Computer Science, for this term.

managers and their employees). This can create a need to support awareness with respect to certain group-behavior interaction styles.

3.3.1 *Asynchronous and Synchronous Collaborative Exchange*

The differences between asynchronous and synchronous collaborative exchange can be summarized by the following time-space matrix [Ellis, et. al., 1991]:

	Same Time	**Different Times**
Same Place	**face-to-face interaction**	**asynchronous interaction**
Different Places	**synchronous distributed interaction**	**asynchronous distributed interaction**

Figure 5: Groupware Time-Space Matrix from Ellis, et. al., 1991.

Note, for the purposes of this discussion, that the *place* is a shared virtual environment. Face-to-face interaction refers to "faces" of avatars. Similarly, synchronous distributed interaction might mean 2 people working in different rooms of a large CVE, at the same time.

Synchronous Collaboration and Awareness

Synchronous collaboration is interaction which occurs while others concurrently share your environment. Synchronous activities do not have to occur at a shared location, even in virtual environments, but they do presume concurrent presence in *some* part of the environment and concurrent participation among collaborators. This behavior is represented in the left side of the matrix in Figure 5. Simple chat programs can be thought of as synchronous software devices, even though there is usually are slight pauses resulting from typing and transmission times. More common examples include: tele-conferencing, video conferencing, and of course, face-to-face communication.

In Tele-immersion, the two cells on the left side of the matrix in Figure 5, above, do *not* refer to the fact that collaborators might be *physically* distributed. Rather, in a CVE, collaborators either engage in avatar-to-avatar (face-to-face) synchronous exchange, or they work simultaneously in *virtually* distributed locations of the VE. As a contrast, the upper right corner of the figure, denoting same place and different time, means that collaborators are in the same part of a virtual environment at different times. This concept leads to the idea of *persistent* virtual environments, which are discussed later.

Tele-immersion has added support for face-to-face communication that some other systems lack. Though video-conferencing effectively conveys most human gestural cues, many other collaborative systems have no such support. Almost like sign-language, people have compensated by adopting various forms of gestural expression. Common examples for chat rooms include the following:

smiling	: ^)
frowning	: ^ (
winking	; ^)

These have the effect of conveying meaning through gesture, which is a form of awareness. With the help of tracking devices, 3D avatars can be animated to include physical gestures like pointing, waving, and nodding. Surely avatars will one day include smiley, frowning and winking.

Asynchronous Collaboration and Awareness

Asynchronous collaborative exchange is peer-to-peer interaction that does not occur concurrently. It is represented in the right side of the matrix in Figure 5. Asynchronous exchange implies that there is a delay of unknown length between one person's actions and another's. There are many common examples of this in modern practice. Email and voice-mail are primary examples of a asynchronous collaborative media. Asynchronous activities and exchanges can take place during the course of more complex activities as well. In shared editors, for example, users may wish to retrieve and edit data that others have previously edited and saved [Ellis, 1991]. Software exists that not only enables asynchronous collaboration, but also manages it. Version control software and meeting and reservation systems are examples of such applications.

In Tele-immersion, awareness becomes important in asynchronous collaboration when changes in the state of the environment occur between one person's visit and another's. In scenario 2.3.2, the user leaves persistent "tags" in the environment to indicated the places where he made changes. Regardless of the form, awareness of change should help reduce an individual's confusion between asynchronous sessions. This may be especially true as *persistent* virtual environments evolve.

Persistent Tele-immersion

The existence of *persistent* virtual environments is another much desired, but seldom realized goal for Tele-immersion [Leigh, et. al., 1997]. Persistent virtual environments are worlds which continue to exist, and in some proposed cases *evolve*, even while they are un-inhabited. In the popular, multi-player, internet game, Quake™, hundreds of servers are registered with central systems and remain online even when no one is playing. As long as the server's owner maintains his or her connection, the environment is persistent. To date, few persistent virtual environments exist for Tele-immersion. One notable example is the NICE project, which has been online for over 3 years [URL 9].

3.3.2 *Navigation and Movement*

Navigation and movement in Tele-immersion are not inherently complex. The words "up", "down", "forward", and "backward" ring true to their exact meaning in the physical world. Many have argued that Tele-immersion, through gesturing, adds even greater value [Leigh, et. al. 1997]. For instance, one can now point while saying, "go over there." The most notable difference between the actual and the virtual is with movement. "Walking" in Tele-immersion is not typically done on one's own two feet. Some research facilities have experimented with motion platform machinery (*e.g.*, treadmill), but traversing an environment in Tele-immersion is usually accomplished through the wand. This happens either by pressing the buttons and/or thumb-pad, or by pointing the wand in the intended direction of travel. The circumstances are slightly more natural for "flying", especially when using a joystick. Real-world flight usually involves joysticks, so little translation is involved when flying in a VE. Beyond that, movement in Tele-immersion requires significant efforts on the part of the designer to engineer a software and *hardware* solution.

An important, perhaps obvious, consequence of the kinds of artificial motion described above is that the user does not actually move in the environment. Rather, the *environment* moves according to the user's steering action. All of the above withstanding, movement and navigation in Tele-immersion are not major problem areas. Any artificiality in the way one moves and navigates or the way the world moves in response to steering was perhaps negated the first time someone experienced motion sickness.

Navigation, Collaboration, and Awareness

Just imagine that you are working in a CVE which models a large factory of several hundred rooms and miles of floor space. Now imagine that you and your colleagues are on opposite sides of the factory, separated by a mile of virtual space, and someone says, "come over here for a minute." Walking just will not suffice; it could take several minutes, even hours. Flying may be sufficient in some cases, but across large distances, *teleporting* seems the most efficient mode of travel.

It is both desirable, and easy, to implement improved means for "traveling" through a large-scale CVE. One or two lines of code are enough to translate a position matrix from one place to another in an instant. The state change that takes place with such a translation is stored in a simple floating point array with three indices. Notably, the cost of broadcasting this information to multiple participants is negligible. One way to take advantage of these facts and move around in the world is through *teleports*, or some similar hypernavigation.

Teleports can be thought of as the hyperlinks for Tele-immersion. They are a basic means for the user to quickly jump from one position in a VE to another. Even before Tele-immersion, teleports were used in multi-player computer games like Quake™. Teleports and doorways were used by both the ArtWorld [Bresnahan, et. al.] and Virtual Atrium [Leigh and Johnson] exhibits at Supercomputing '98. The VT-CAVE's contribution to the atrium included teleports, but notable issues of awareness were immediately apparent to early users at EVL.

Usability and Hypernavigation

One of the first things we learned about the usability of hypernavigation was that portals need to be visibly and meaningfully marked. Early VT-CAVE teleports tended to blend with the background. Citing also the sudden, unannounced change in position, the EVL team in Chicago reported teleportation to be confusing. One person even remarked that he thought the program had crashed or otherwise malfunctioned. This comment, and others like it, seemed to confirm that the transition from one point to another should be smooth and perhaps even be signaled by an alert. In ArtWorld, transition between locations is relatively smooth; it has a deliberate delay. (Unfortunately, the extremely poor network conditions on the SC '98 conference floor made it difficult to determine the true duration of transition at the time.) ArtWorld's doorways are also well marked (*i.e.*, they look like doors), but the "signs" identifying each door are highly abstract. Symbolic icons are used with the intent of graphically relating something about what is on the other side of the door. Given the rather abstract nature of the entire ArtWorld environment, it is questionable whether or not the icons are effective. At SC '98, portals required explanation on more than one occasion.

At the time, the Virtual Atrium was the best example of well-designed hypernavigation. Portals are plainly marked, and in fact, are the only real items of interest in the main room. Signs are posted outside each portal with an image and description of what is on the other side of the door. Upon entering the portal, which is more like an elevator, the sound of a harp being strummed is heard and the world moves with the effect of being lifted far above the ground. For some added efficiency, the transitional delay was slightly exaggerated, allowing time for the new environment to be loaded.

Dynamic Hypernavigation

Each of the examples of hypernavigation mentioned above is the same in one way. They all place teleports at fixed and unchanging locations throughout the environment. An alternative method is to allow the user to open doors dynamically from anywhere to anywhere. The user references the point of destination by absolute world coordinate or in relation to some object or other collaborator in the world. Provided this action is not easily invoked by mistake, this method should eliminate any confusion about when and to where one is teleporting. More importantly, the flexibility and range of hyper-navigation is extended tremendously.

Again, a dynamic teleport should include some support for awareness. Particularly, if the port is being opened to the location of another participant, the system might require the initiator to ask for permission before moving into the other person's personal space.[8] The Collaborative Console supports this kind of awareness by requiring any teleport*er* to wait for confirmation from the teleport*ee*. The teleportee is made aware of the incoming teleporter when he receives an incoming teleport request. The teleport function then waits for the teleportee to reply either "YES" or "NO". If confirmation is denied or otherwise not received within some time-out period, then the teleporter is made aware of this fact via some feedback from the system.

Providing Navigational Awareness to Others

The previous treatment of navigational awareness is almost entirely in the context of providing feedback to the user invoking the action. The section on dynamic hypernavigation, however, begins to suggest that certain navigational modes require some level of mutual awareness toward the impending action(s) (*e.g.*, the request and confirmation exchange for certain teleport actions). But even when the teleport action does not require the active participation of a remote user, that remote user still requires some level of notification to avoid confusion. This was observed at VT-CAVE when one person would randomly, and without notification, teleport out of view of another. The response from anyone witnessing this for the first time was that they supposed the remote machine had crashed. In actuality, the first person only walked through a teleport.

Like most things, each of these methods will serve its purpose at appropriate times. Scenarios involving commonly visited locations, or which require users to visit specific locations, might require fixed ports. On the other hand, scientists working freely in a large design environment might wish to jump to a colleague's location "on the fly". This latter example raises some other issues of collaborative awareness which might be tempting to overlook; privacy and etiquette. For example, a person momentarily working alone in a CVE might have reason to prevent others from arbitrarily entering his virtual space. In any case, it would seem reasonable for users to somehow broadcast intent before hypernavigating into another person's virtual space.

3.2.3 *Social Interaction and Group Behavior*

Issues of appropriate social interaction and behavior are sure to arise in Tele-immersion just as they do in personal interaction and other forms of computer-mediated collaboration. Social protocols exist that

[8] Perhaps more specifically, the collaborative application should have a *policy* governing this type of awareness. Whether or not confirmation is required depends upon the scenario of use. It is assumed that when people are cooperating, they would use a confirmation policy.

dictate organizational structure, group process, shared control, and the like. Each of these items requires support for awareness [Begole, 1998, p. 15].

For the most part, traditional groupware has been forced to rely on artificial cues to represent social behaviors like turn-taking. Consider a shared whiteboard, for example; some indication must be made that shows who "controls the floor". This could just be the user's name displayed as a banner at the top of the screen. Tele-immersion, however, is a medium which aspires to reduce these artificial means, whereby notions of appropriate behavior are assumed much more naturally. Continuing with the whiteboard example, an avatar standing at the board and writing on it will be an obvious indication of floor control.

Attention awareness is another prime example of how Tele-immersion might provide natural indications for gauging social interaction. With a simple glance, one might observe two colleagues': their avatars standing face-to-face, hands and heads moving about, and faint sounds emanating from their direction. A rationale conclusion, and one which requires no artificial support, is that they are engaged in a conversation and perhaps should not be disturbed.

Privacy

The Spatial Model of Interaction seems to allow for the implementation of privacy through its focus and nimbus components. A nimbus sized so small as to seem non-existent makes it possible to eliminate any awareness others have toward that nimbus, no matter what their focus. Further, focus can be directed toward only certain nimbi and through restricted channels. This makes it possible to reduce distractions from others in the environment. If a person is maintaining privacy, however, anyone trying to contact or interact with that person ought to be notified of the situation so as to avoid confusion.

Chapter Four:
The CAVE Collaborative Console

This chapter contains the detailed description of software being designed and implemented to support collaborative awareness and interaction in Tele-immersion. The application, called the CAVE Collaborative Console (CCC, Console), is still under development at VT-CAVE, but is freely available upon release of this paper.

4.1 Bringing Collaboration into the CAVE

The CAVE Collaborative Console extends CAVERNsoft and LIMBO by providing users with a suite of prototypical interface devices and functions that are believed to be useful in many types of collaborative sessions. The application programming interface (API) is intended to be somewhat generic and flexible and yet is implemented with many utilities that support awareness, presence, and collaborative manipulation. It is recognized that there are many uses for Tele-immersion. But, it is believed that many users will require at least a base set of collaborative utilities and tools to cooperate with multiple participants in most virtual environments. By designing a flexible interface, the Console can be tailored toward specific domains of use through minimal end-user programming. The Console is equipped to integrate both existing tools and tools which were created at the VT-CAVE.

4.2 Object-Oriented Design

The API for the Collaborative Console was designed according to the Object-Oriented Paradigm. Relying on CAVERNsoft for network communication and database management, Console classes provide content by way of objects and collaborative utilities.

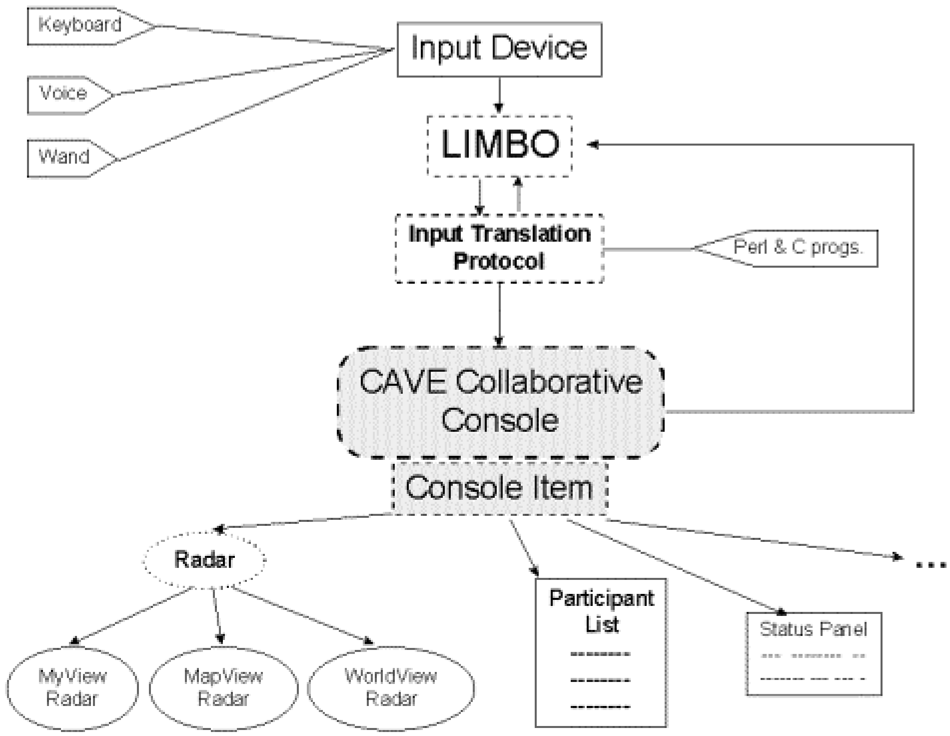


Figure 6: Object View of the CAVE Collaborative Console

4.2.1 *Component Model and Description*

This section covers the components of the console, including LIMBO, and explains their relationship to one another according the figure above. The "Input Device" node of the model in Figure 6 has three flags, "keyboard", "voice", and "wand", which indicate each mode of input recognized by the Console. The remainder of this explanation will refer to the voice-activated input. All input is initially retrieved by calls from within LIMBO, represented by the second node in the model above. This input, however, is immediately passed along to the "Input Translation Protocol" (ITP), which parses input for legal Console commands. The flag labeled "Perl & C progs." indicates that separate Perl and C programs are initialized and run to handle the parsing (see section 4.3.1). The ITP passes these commands and any arguments they require to the Console. Finally, the Console invokes the action specified by the command and makes the appropriate updates to the LIMBO environment.

Programming Environment

The remainder of this discussion assumes a programming environment which includes C++ and Iris Performer. The `Console` object is not derived from any Performer object, but individual `ConsoleItems` are. A generic `ConsoleItem` is derived from the Performer class `pfDCS`. In the current version (v.0.5), there are three sub-classes which derive from `class ConsoleItem.` `They` are: Radar, ConsoleList and ConsolePanel. Each is derived from the ConsoleItem class and inherits attributes from both ConsoleItem and pfDCS. Any future interface devices will inherit the ConsoleItem class.

class Console

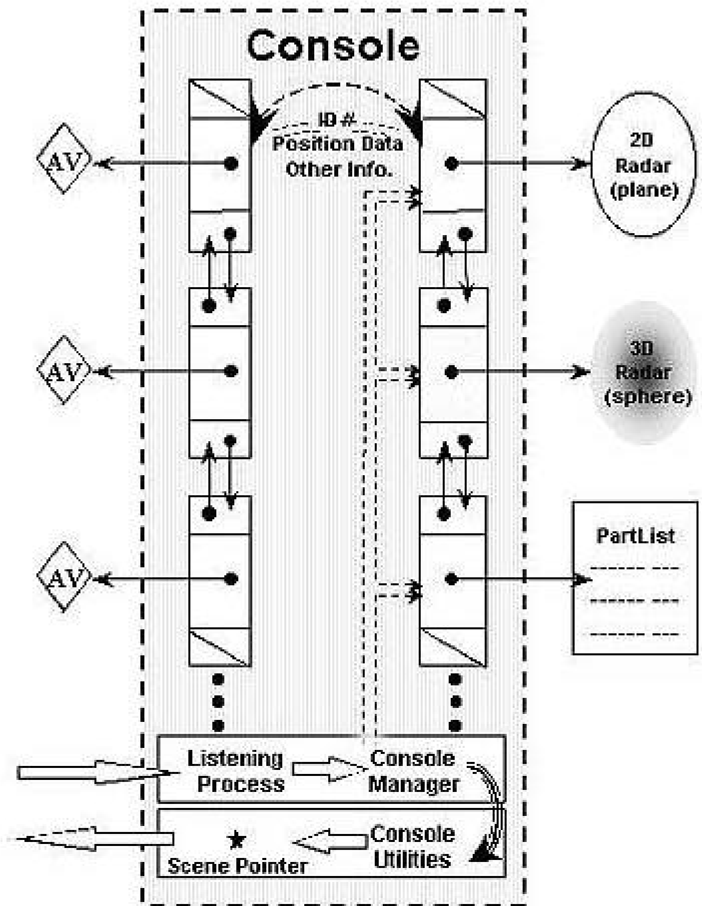


Figure 7: Model of the Console as a "container" of Lists of ConsoleItems

The Console class is a container of ConsoleObjects as well message handler for acting upon those objects. Figures 7 shows how the "containers" are actually doubly-linked List templates for which ConsoleObjects are the parameterized type. The List template can, of course, hold objects of any data type. Currently, separate lists are used to hold objects that are either of type ConsoleItem or type CAVERNst_baseAvatar_c. Two List objects are instantiated by the console, one for each type of ConsoleObject.

One list, called current_users, holds pointers into the avatar database manager in LIMBO. Each node in the current_user list points to a different CAVERNst_baseAvatar_c object. Avatar objects can then be queried for data like ID or position. This information is maintained within the `Console` so that the individual `ConsoleItems` can have quick and easy access to this data.[9]

The second list, called `current_items`, holds pointers to the items which were installed when the `Console` was initialized. Each node in the `current_items` list points to a different `ConsoleItem` object.

class List <class ConsoleObject>

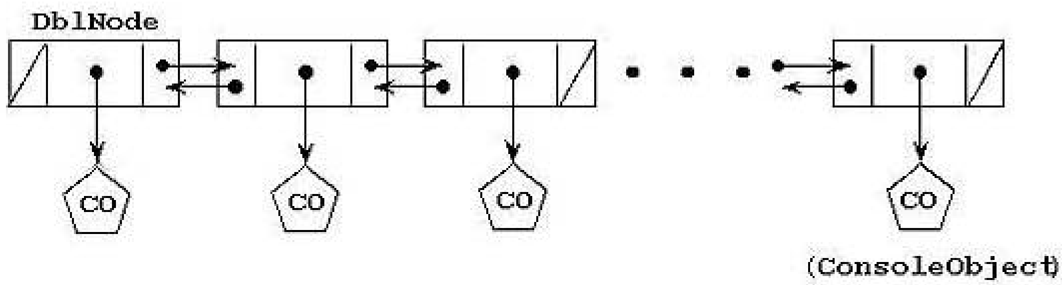


Figure 8: Model of the List class

The `List` class is a generic template class for holding `ConsoleObjects`. Nodes on the list (`ConsoleObjects`) are doubly-linked for ease of manipulation. Two `List` objects are used to hold `ConsoleObjects` of either type `ConsoleItem` or type `CAVERNst_baseAvatar_c`.

class ConsoleItem : public pfDCS

The `ConsoleItem` class is the parent class for all objects on the Console. In the current version (v0.5), three main sub-classes are derived from the `ConsoleItem` class. They are: `Radar`, `ConsoleList`, and Console`Panel`, and are explained in further detail below.

All `ConsoleItems` have the following in common:

Data Members

- `ItemType itemType;`
- `int shown;`
- `Inherited pfDCS members`

Functions

- `ItemType getType();`
- `int getViewState();`

[9] This data is also maintained by LIMBO and may be removed from future versions of the Console.

- void setViewState(int vs);
- virtual void addUser(CAVERNst_baseAvatar_c *new_user);
- virtual void removeUser(CAVERNst_baseAvatar_c *old_user);
- Inherited pfDCS functions

class ConsolePanel : public ConsoleItem

The ConsolePanel class is the parent class for certain text-based displays. In the current version, one subclass is derived from the ConsolePanel class; StatusPanel. This class is intended mainly for future versions of the Console which will include graphical user interfaces.

class ConsoleList : public ConsoleItem

The ConsoleList class is the parent class for certain other text-based displays. These are mainly text lists showing the names of remote collaborators or of object files which have been loaded into the environment. In the current version, two subclasses are derived from the ConsoleList class. They are ObjectList and ParticipantList.

All ConsoleLists have the following in common:

Data Members

- pfFont *fnt;
- pfDCS *titleDCS;
- pfText *titleText;
- pfVec3 position_;
- float prevTime_;
- int active_;
- Inherited ConsoleItem and pfDCS Members

Functions

- double getDistance(pfVec3, pfVec3);
- Inherited ConsoleItem and pfDCS Functions

class ObjectList

The ObjectList class provides the interface for a heads-up text list of objects that users have loaded into the environment (or any other object of which the user wants to keep track). Each object is identified by its name and its distance and direction (North, South, East, West) relative to the local user. Objects on the list include models which are loaded in using the load command, or any other objects the user wishes to identify.

`class ParticipantList`

The `ParticipantList` class provides the interface for a heads-up text list of all collaborators sharing the environment. Each person is identified on the list by name, distance and direction from the local user. Remote participants are added to the list as soon as they enter the environment. This is done through the addUser function, which is overridden from class ConsoleItem.

`class Radar : public ConsoleItem`

The `Radar` class is the parent class for the specific kinds of radar available to the user. In the current version, four subclasses are derived from the `Radar` class. The are: `ThreeSpaceRadar`, `FloorPlanRadar`, `My_tsRadar`, and `My_fpRadar`.

All Radar objects have the following in common:

Data Members

- `pfVec3 center_coords;`
- `char objName[256];`
- `pfGroup *blips;`
- `pfGeode *radar;`
- `pfGeode *center;`
- `Inherited ConsoleItem and pfDCS Members`

Functions

- `void setCenter(pfGeode* ctr);`
- `void setCenterCoords(pfVec3 ctr_coords);`
- `Inherited ConsoleItem and pfDCS Functions`

`class ThreeSpaceRadar : public Radar`

The `ThreeSpaceRadar` class models the radar as a view of the world mapped to a sphere. The center of the radar can be specified by the user to be any object or arbitrary 3D coordinate. The default center values are (0,0,0), but these are merely default settings and may or may not correspond to the center of the virtual world. All objects and users, including the local user, are drawn relative to whatever the users sets as the center.

`class FloorPlanRadar : public Radar`

The `FloorPlanRadar` class models the radar from a bird's-eye view of the world mapped to a circular plane. The center of the radar can be specified by the user to be any object or arbitrary 2D coordinate (the third dimension is always zero (0)). The default center is (0,0,0). All objects and users, including the local user, are drawn relative to this point.

class My_tsRadar : public Radar

The My_tsRadar class creates a version of the ThreeSpaceRadar which is centered on the local user. All other remote collaborators and objects in the shared environment are positioned on the radar relative to the location of the local user.

class My_fpRadar : public Radar

The My_fpRadar class creates a version of the FloorPlanRadar which is centered on the local user. All other remote collaborators and objects in the shared environment are positioned on the radar relative to the location of the local user.

Table 1: Classes in the CAVE Collaborative Console

Class	Inherits	Description
Console		Manages the list of ConsoleItem objects that are initialized with consoleInit(int argc, int argv[ ], pfScene *). The Console also keeps a list of pointers into LIMBO's avatar database for communicating changes in the avatars' states to ConsoleItem objects that use this information
List<class ConsoleObject>		Doubly-linked list for holding pointers to ConsoleItem objects or CAVERNst_baseAvatar_c objects
ConsoleItem	pfDCS	Generic parent class
Radar	ConsoleItem	Generic parent class
ConsoleList	ConsoleItem	Generic parent class
ConsolePanel	ConsoleItem	Generic parent class
My_tsRadar	Radar	3D (spherical) radar whose center is the *local user*
My_fpRadar	Radar	2D (planar) radar whose center is the *local user*
ThreeSpaceRadar	Radar	3D (spherical) radar whose center is *some location in the environment*
FloorPlanRadar	Radar	2D (planar) radar whose center is *some location in the environment*
ParticipantList	ConsoleList	Textual heads-up display of all *users* being tracked by the Console
ObjectList	ConsoleList	Textual heads-up display of all *objects* being tracked by the Console
StatusPanel	ConsolePanel	Textual heads-up display of status information about any collaborator (including self). Status information might include: localTime, connectTime, or other personal information

4.3 Functional Description

This section covers the Console's collaborative functions and utilities from a user's perspective. It is highly desirable that all collaborative utilities be accessible while the user is still immersed. At the beginning of this research very few collaborative tasks could be performed that did not require the user to return to the terminal. By including support for voice recognition input, most Console features can be invoked while immersed.

Collaboration Aware Software

The CAVE Collaborative Console is a c*ollaboration aware* application [Begole, 1998]. Ironically, the use of the word "aware" is coincidental and has nothing to do with awareness as used in this paper. A collaboration aware application means that it was designed specifically for a multi-user environment. The distinction from a single-user environment seems obvious, but the particular impact of the collaboration aware design is indirectly noted in Leigh, et. al., [1997] and is reiterated here. Leigh, et. al., claim, and local experience supports that attempting to "retro-fit" a single-user Tele-immersive application into a multi-user application is not the preferred method [p. 6 of 9].

4.3.1 *Voice-Activated Input*

The CAVE Collaborative Console was designed to allow Tele-Immersion applications developers to integrate their own input devices. The default input device (the wand) is best used as a point-and-click, grab-drag-and-drop, or steering device. Because the focus of this research is on collaborative functionality, rather than usable interface design, it was decided that the development of GUIs for the wand was beyond the research scope. Voice-activated input, though undocumented, is used by some members of the CAVERNUS community. With the assistance of Stuart Levy from NCSA, a voice-activated protocol was adopted and implemented for the Console [URL 11].

The voice-activated input involves the use of a separate, commercial off-the-shelf (COTS) application for the PC called *Dragon Naturally Speaking*. *Naturally Speaking* translates words spoken into the PC's headset microphone into text. The text is typed into the active window on the PC's monitor. In the case of the Console, the active PC window is a telnet session to the CAVE machine. On the UNIX (IRIX) side of this protocol, a Perl program called *throw* parses the text for legal Console commands and a C program called *catch* provides 3 simple routines for opening a socket, retrieving the parsed command, and closing the socket.[10]

[10] *throw* and *catch* were written by John Kelso, Senior Research Associate, VT-CAVE. See "Acknowledgements" for further details.

throw

`throw` is a Perl program which handles the parsing of text typed into the telnet window. Command syntax for `throw` is defined in a parsefile called `grammar`. The `grammar` file first defines the command delimiters, `$start` and `$stop`. The default delimiters are "computer" and "please", respectively, and can be easily changed. All text typed outside the command delimiters is ignored, while all text that appears between the start and stop words is tested against the set of defined command syntax. Words that match legal commands are placed in a buffer which is read by the C routine `catchData(char line[size], size)`. Words which do not match can either be logged in a text file or discarded. This has the effect of not requiring the user to switch between command input and normal conversation. At the same time, it is not clear whether or not words spoken by a user on one end will be picked up by the recognition system of the user on the other end.

catch

`catch` is a C program which provides three simple routines for opening a socket to `throw`, retrieving commands, and closing the socket. These three routines added to the Console's source code are as follows:

- `void catchInit() -` Opens a socket with the telnet window running `throw`.
- `int catchData(char line[size], size) -` Returns up to `size` bytes of data in `line` or zero (0) if no data has been sent.
- `void termCatch() -` Closes the socket with `throw`.

4.3.2 *Console Objects*

The CCC was also designed to be somewhat flexible on the component side of the interface. The only constraint is that new `ConsoleObjects` be Performer-based objects.

Awareness Tools

`(FloorPlanRadar, ThreeSpaceRadar, My_fpRadar, My_tsRadar, ShareView , ParticipantList, StatusPanel)`

`Participant List` - a heads-up list of everyone sharing the environment. Other collaborators are displayed by name and distance. For example, "Mary: 25.5" would indicate that Mary shares the environment and that she is 25.5 units away.

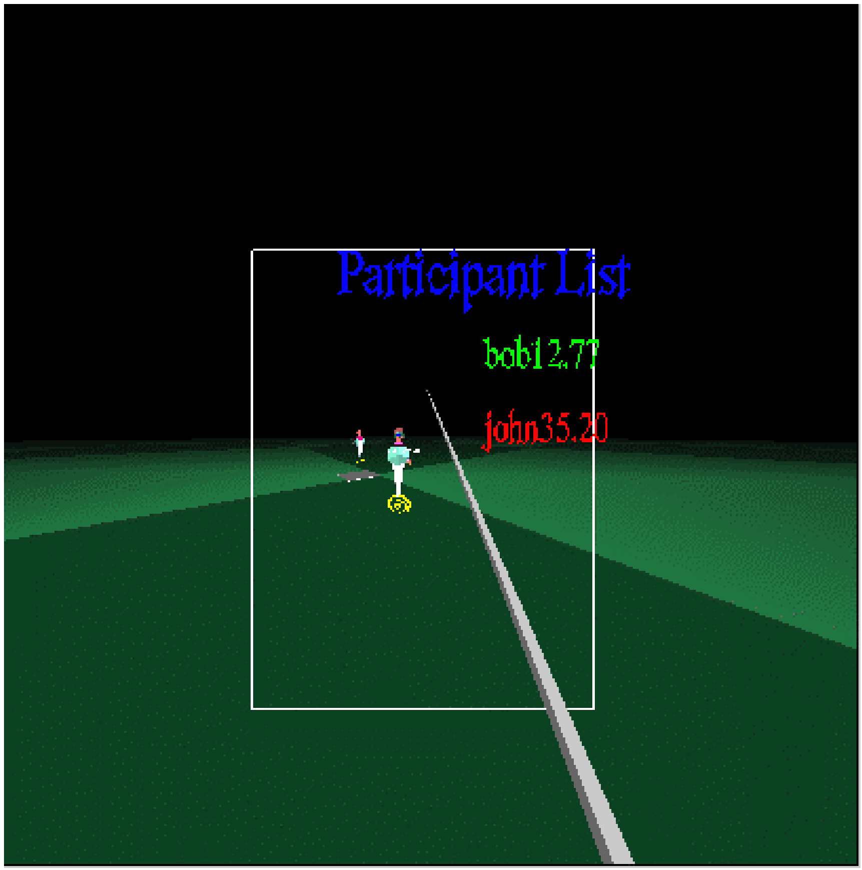


Figure 9: Simulator Screen-shot of Participant List Showing Two Other Participants

`FloorPlanRadar, My_fpRadar` - allow the local user to track remote participants and other objects in two dimensions. This is more appropriate for collaborative sessions which do not involve "flying". In architectural walk-throughs, for example, users typically navigate while on the ground.

Figure 10: Simulator Screen-shot of My_fpRadar Showing 2 blips

`ThreeSpaceRadar, My_tsRadar` - allow the local user to track remote participants and other objects in three dimensions. This is more appropriate for collaborative sessions which involve "flying". By capturing the z-axis, the spherical model relates the space above and below the radar's center.

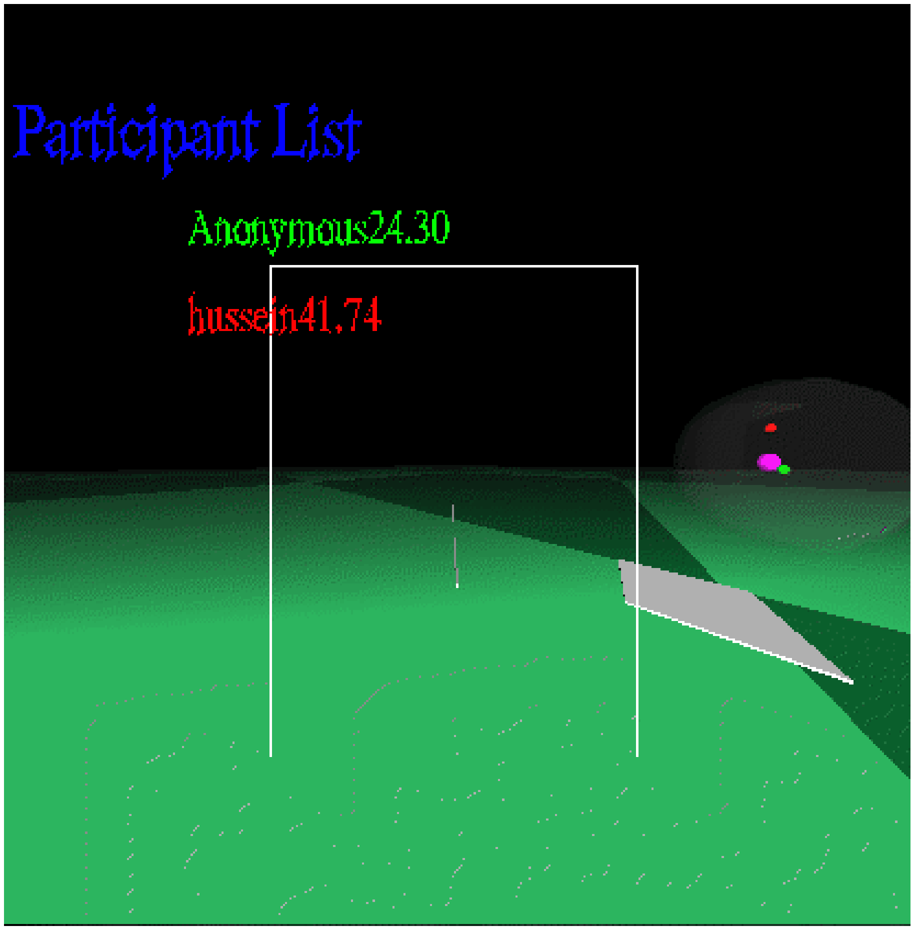


Figure 11: Simulator Screen-shot of ThreeSpace Radar Showing 2 blips

4.3.3 *Console Utilities*

Awareness Methods

`highlight` - Indicate an object through some means of increasing the user's ability to perceive it. Depending on the object, this is analogous to highlighting the user's name on a list by changing its color or drawing a box around some geometric object. When the object parameter is the name of another participant, all active awareness devices, including avatars[11], invoke their particular instances of this function. For example, if a radar and the participant list are both active on the console, invoking the **`highlight <user>`** command will uniquely identify that user's name in the list, as well as uniquely indicating that user's position on the radar. The highlight command can also be invoked by the console in conjunction with other functions. When sharing another's view, for example, it might be helpful to be visually reminded of who's view you are sharing.

shareView - Simultaneously share another participant's or object's view, while your avatar remains at your original location. The object's or participant's view being shared is displayed as text on the CAVE's master wall. If a name appears on the list and/or a blip appears on the radar, that indicator is uniquely identified as the object or participant sharing the view.

show/hide - Toggle the appearance of any Console object. Anything revealed by **show** can be removed from view using **hide**.

view - View a specified object or absolute world location from a specified **camera_position**. **camera_position** can be: **CAM_FRONT | CAM_BACK | CAM_LEFT | CAM_RIGHT | CAM_TOP | CAM_BOTTOM**. The **camera_position** is displayed as text on the CAVE's master wall. If the object being viewed is listed on the Participant List and/or shown in the radar, those indicators are uniquely distinguished from other objects.

Navigational Utility Methods

Navigational utilities are a unique form of collaborative support. Because of the tendency in Tele-immersion for environments to become large and complex, it can be helpful having easy ways to move quickly to another person's location in the world. Certain "steering" utilities can be of further use to guide another user, perhaps a trainee, along a preset path.

savePath / trace - Record position coordinates while navigating through the environment. These coordinates are written to a file and can be used to recreate the navigation with a call to **loadPath(x,y,z)**. Coordinates are stored in a **.path** file and read in by the Console. (NOTE: It is likely that only some points will be save and the rest extrapolated. The system could potentially save millions of points in a matter of seconds.)

teleport - (jump | move | port | translate) - Automatically translate your position to any desired **location**. The location follows the command word and can be in the form of a participant's or object's name or absolute world coordinates.

tether (attach | follow) **/detach** - Bind your location relative to another participant or object at a specified distance. The actual location of your avatar and your view will be behind and to the right. The participant or object to whom you are attached will be displayed as text on the CAVE's master wall.

---

[11] The avatar themselves do not invoke this function. Some method in the console invokes it to indicate

All active awareness devices will uniquely distinguish that participant or object from other objects and participants. A `tether` is indicated by a **`cord`** that attaches you to the participant or object. The length of the `cord` can be changed using **`adjust`**. A `tether` is released using **`detach`**.

`adjust` - Change the length of a tether `cord` to a specified `length`.

The nature of the `cord` is not rigid. The attached user is free to move within the confines of the half-sphere whose radius is created by `cord->length.`

`tracePath` - Follow a pre-defined **`path`** at a **`rate`** whose value is set by the user when invoking the command. This could be like tracing over a colleague's actions in an asynchronous collaboration, or following the path of a ship in orbit around a distant planet. The distinction from the **`teleport`** command is that this method is not simply point-to-point, but rather along a defined path. It is to be assumed that a **`path`** has been saved into a file which can be read back into the program. This is done using the **`savePath`** command (above).

Collaborative Exchange and Discovery Recording Utilities

One limiting factor in this area, that is currently being addressed by VT-CAVE and others in the CAVERNUS community, is the need for feedback from the steps involved with each of these procedures. One solution suggested has been to instantiate another X window on the CAVE wall and pass voice input to the window. This way, the user can see what was being typed as a result of his or her dictation. This might be helpful in drafting emails, for example. Here again, usability testing is needed to determine if this is useful.

`voiceMail` - Dictate message for an asynchronous, verbal exchange with one or more collaborators. This utility is one way of leaving reference "tags" in a persistent virtual environment.

`mediaCapture` - Screen-capture the simulator window as a digital movie file of the session using the SGI Media Recorder. This method captures an egocentric perspective of the immersive experience as if looking through the user's glasses. This method is best suited for scenarios which are meant to describe and display an experience but it is somewhat limited by the amount of disk space it requires. In our experience of recording just a few minutes, the recording process saved an extremely large file, which, when even when compressed, was still several megabytes in size. For recording critical data this is

---

an avatar's actual position in the world.

acceptable, but this would not be a good function to invoke for the purpose of recording "meeting minutes."

`videoTape` **-** Record the session on VHS tape. Video feeds in from either the simulator window or a outside camera filming the person standing in the CAVE. This method is somewhat more flexible and less of a strain on disk space than the Media Recorder. This is the function which is best suited for saving the entire course of a virtual session. In fact, we have used this feature to send visitors home with a VHS tape of their visit. **VIDEO TAPE** is, however, one of the only features of collaborative interaction which cannot currently be invoked by the console nor while immersed. Rather, the user starts the recorder before entering the immersive environment. **VIDEO TAPE** can record the egocentric view described above, or it can record an outside camera view of the person standing in the CAVE. This view might be best suited for usability testing; to see what the user is doing physically. Video captured from the simulator window is saved directly to the VHS tape and does not consume disk space.

Chapter Five:
Conclusions and Comments on Future Work

This paper represents a careful examination of collaborative awareness, particularly in the context of Tele-immersion. Prior to this treatment, others created the infrastructure which makes computer-mediated interaction feasible over Tele-immersive networks. Important work was also underway to identify and support certain kinds of collaborative awareness and basic interaction. This thesis continues upon these foundations in two significant ways:

1) This research, supported by both literature survey and local discovery, offers a greater understanding of the impact of awareness on collaborative interaction.
2) Software developed in conjunction with the research adds much needed utility to the software base for the CAVE™ and CAVE™ networks (including the Immersa-Desk™ and CAVE™ Simulator).

Success in the laboratory environment is gaining attention and moving into government and industry. Yet, Tele-immersion is a relatively new concept and still aspires to higher levels of usability. The remainder of this chapter summarizes the results and contributions after one year's worth of research. It discusses the significance of these results and offers comments and suggestions for future work.

5.1 Summary of Results

This thesis offers the following contributions:

1. *Survey and examination of collaborative awareness in immersive virtual environments*

   The earliest achievements supporting networked CVEs and Tele-immersion were, understandably, focused on issues surrounding performance - latency, lag, jitter, and shared data fidelity, for examples. Some advances were also made with regard to the most elementary components of human-to-human interaction in networked virtual environments. Still, no comprehensive analysis of collaborative awareness in immersive VEs existed. This treatment brings together the main themes and key concepts of its predecessors, while offering new insight based on new experiences.

2. *The quality of Tele-immersive collaboration can be improved by recognizing a connection with CSCW and early CVEs and extending their early discoveries*

   When the staff at VT-CAVE began experimenting, Tele-immersion was a technology only a few years old. The first stages of research in Tele-immersion included attention to just a few details of collaborative awareness. The work of Leigh, et. al. was extremely beneficial at this point. Still, much more is known about collaboration via *groupware*. Armed with this knowledge base and significant contributions from Benford, Bowers, Greenhalgh, et. al. in the area of CVEs, we at VT-CAVE are able to create software acutely focused on improving collaborative interaction in Tele-immersion.

3. *Basic user requirements for collaborative awareness and interaction in Tele-immersion can be easily and efficiently supported*

   It was discovered during the course of designing and implementing the CAVE Collaborative Console that many features and basic requirements can be supported easily and with very little overhead. The CAVERNsoft/LIMBO foundation on which the Console is built had been thoroughly tested for usability based on system and network performance. The extensions made by the Console involve the use of rather lightweight objects and simple operations on those objects.

4. *An Tele-immersive application can be designed to support a flexible range of input devices and a easily customizable interface*

   Because Tele-immersion is still very much in its early growth stages, change occurs often. With this in mind, the Console was designed to support the integration of many different input devices. Similarly, the current `ConsoleItems` are considered to be prototypes for more specialized interface devices. These devices can be easily added to the Console.

5. *A new system can be integrated with existing support systems while requiring only the minimum expected changes to these support systems*

   It was realized early in the design of the CAVE Collaborative Console that any new system based on CAVERNsoft and/or LIMBO should impose a very few number of changes in the existing code. It was clear, through comments in the CAVERNsoft and LIMBO source code, that certain changes are expected. These include: expanding the types of information stored in the avatar database, customizing communication channels and registering new callbacks for said channels, and customizing policies governing shared and replicated data. The Console makes use of these "hooks", but requires those wishing to deploy the Console over an existing CAVERNsoft or LIMBO application to make no further changes to either CAVERNsoft or LIMBO.

5.2 Comments on Future Work

The Console continues to be developed beyond the scope of this paper. Throughout the course of this research and the development of the Console, it has been recognized that awareness is only one component of collaborative interaction. While clearly fundamental, awareness is not concerned with some of the more practical utilities of collaboration over computer networks. In an attempt to deal with the numerous demands and subtleties of supporting interaction in Tele-immersion, this author has assembled a suggested list of requirements. The list is not purported to be either comprehensive or definitive. It is merely the suggestion of some future framework which might make the development process easier.

Applications which intend to support collaborative interaction in Tele-immersion should address the following concerns:

1. *How to provide participants with awareness attention, action, and location and in a shared environment* -

   This concerns the degree to which multiple participants are able to locate their colleagues and possibly determine their colleagues' actions and focus of attention [Greenhalgh and Benford, 1997, Roseman and Greenberg, 1996, p. 67, Benford, et. al., 1994]. In JAMM, multi-user applets show where different users are working within a single shared document [Begole, 1998]. In large-scale, multi-room CVEs, it is also important that users have ways of finding each other [Curry, 1999].

2. *How to establish a user's sense of presence and awareness of the environment* -

   This involves the degrees to which you feel that the environment exists and that others sharing the environment with you [Benford, et. al., 1996, pp. 78, 79]. It includes support for perceiving changes made to the environment by remote participants. To a variable degree, presence and environmental awareness are expressed in all kinds of collaborative environments, from chat rooms to immersive CVEs.

3. *How to support personal, shared, and multiple views* -

   This is the ability for individuals to share views with others (WISIWYS), or to control their own unique view of data and the environment [Begole, 1998, Snowdon and Jää-Aro, 1997]. Begole has noted that as early as 1987, people were realizing the restrictive nature of what he also calls "tightly coupled collaboration" [1998, p. 12].

4. *How to support collaboration through synchronous and asynchronous communication* -

   Collaborators work together in one of four modes of interaction: same time / same place, same time / different place, different time / same place, and different time / different place [Ellis, et. al., 1991, p. 41]. Collaboration across time zones, as well as the nature of our work behavior make asynchronous interaction a common occurrence.

5. *How to include participants with slightly different hardware configurations* -

   This handles heterogeneity among users systems. This is important for ensuring that the fewest number of people are excluded from participating due to deficiencies of computing resources [Roussos, et. al., 1998, pp. 6-7 of 17].

6. *How to record and save collaborative sessions for future reference* -
   The methods for saving and recalling data at a later time are important asynchronous activity and occur routinely in work situations. The most common example is the basic task of saving, but in groupware, it is also possible to leave "tags" in the environment to which other users will refer for update information. These tags can assume many forms: links to messages, edit markings, recordings of a procedure, etc. [Leigh, et. al. 1997, p. 219]. Finally, one may wish to record the entire collaborative session. At the VT-CAVE, this can be done with a common VHS recorder and camera. The VHS tape can capture either the straight camera view of the user inside the CAVE, or the CAVE Simulator view of the environment as it is seen by the user.

7. *How to maintain persistent environments and data* -
   This is the ability to save and *maintain* the state of the environment regardless of whether or not the environment is occupied. This is important for asynchronous collaboration across time zones [Leigh, et. al., 1997 p. 224, 1997 p. 2 of 9] and for allowing users to access the CVE at random opportunities.

8. *How to mediate turn taking and control* -
   Shared objects require locking and unlocking mechanisms and other forms of error checking to negotiate manipulation of shared objects. In a virtual environment, these should be transparent to the user so that natural interaction takes place [Leigh, et. al., 1997, Roseman and Greenberg, 1996, pp. 67]. Issues surrounding privilege, security, and integrity of shared data [Leigh, et. al., 1997] frequently and continually come into play. Roseman and Greenberg refer to this as "technical support to deal with multiple distributed processes" [1996, p. 67].

9. *How to maintain Quality of Service (QoS)* -
   This deals with the effect of system performance on usability [Leigh, et. al., 1998, 1997]. This is especially important in long-distance collaboration over computer-networks. In order to minimize latency across networks, models are often distributed to each remote computer. Often, only changes in the model's state are transmitted across the network. One QoS problem often experienced is a *breakdown*, where a remote client crashes but local systems fail to notify the local users, or otherwise "clean up" the remote user's presence from the local machine's environment [Greenhalgh and Benford, 1995, p. 253, Bowers, et. al., 1996, p. 383].

10. *How to use video and audio via channels to allow both public addressing as well as private conversations to occur* -

    This involves any policies for controlling who receives communications from whom. Some scenarios might call for public broadcast and some for private, client-to-client broadcast. This issue can be subtly complex. For example, someone making a public broadcast may still go "unheard" by collaborators who choose to restrict their focus so as not to perceive the broadcast.

11. *How to effectively transmit non-verbal cues*

    This mainly involves the concept of *gesturing*, and represents a significant challenge for most groupware systems. In online communities, like Multi-User Domains (MUDs), it is not uncommon for users to adopt colloquialisms (e.g., using punctuation marks to draw "smiley-faces") or type their gestures (e.g. typing '*shrugs*' to indicate shrugging ones shoulders). In Tele-immersion, the goal is to move away from these artificial forms of gesturing to something more natural, perhaps through motion tracking [Leigh, et. al., 1997, p. 217]. Some systems even strive to incorporate gesture recognition that simultaneously invokes an action while indicating to others that the action is about to take place [Pedersen, et. al., 1993, p. 396].

12. *How to provide a usable interface for collaborative interaction and manipulation that leverages natural metaphors* -

    Depending upon who you ask, natural metaphors are either canonical or cliché in modern software and usability engineering. What is deemed "natural" seems to vary from one design team to the next. Tele-immersion is not an *a priori* source for natural interaction but it does strive to leverage a wider range of interaction styles than traditional groupware systems.

13. *How users navigate through a shared space* -

    This is a basic issue in any kind of shared workspace, but has particular importance in Tele-immersion. Beyond the obvious fact that navigation in Tele-immersion is three-dimensional, a variety of navigation *styles* are emerging. These include: walking, flying, and teleporting. In some cases, users "tether" themselves to one another so that one user can lead the other through the environment.

16. *How to support different kinds of groups with different tasks and needs* -

    This concerns the flexibility of the application design to support a variety of user-types with multiple domains of use [Roseman and Greenberg, 1996, p. 67].

If Tele-immersion and other IPTs are to be successful platforms for collaboration, the research must continue to address fundamental human factors. The current work is meant to provide both an in-depth treatment of collaborative awareness and also motivation for expanding the research base for CSCW via Tele-immersion.

References

Abrams, H., Watsen, K., Zyda, M., "Three-tiered Interest Management for Large-scale Virtual Environments."

Benford, S., Brown, C., Reynard, G., Greenhalgh, C., "Shared Spaces: Transportation, Artificiality, and Spatiality," *CSCW '96: Computer-Supported Cooperative Work*, 1996, pp. 77-86.

Benford, S., Bowers, J., Fahlèn, L., Greenhalgh, C., "Managing Mutual Awareness in Collaborative Virtual Environments," *Proceedings VRST'94*, August 1994.

Bowers, J., O' Brien, J., Pycock, J., "Practically Accomplishing Immersion: Cooperation in and for Virtual Environments," *Computer Supported Cooperative Work '96*, 1996.

Curry, K., "Supporting Collaborative Awareness in Tele-immersion," *Proceeding of the 3rd Annual Immersive Projection Technologies Workshop*, 1999, pp. 253-261.

Ellis, C.A., Gibbs, S.J., Rein, G.L., "Groupware: Some Issues and Experiences," *Communications of the ACM*, Vol. 34, No. 1, January 1991, pp. 39-58.

Finholt, T. A., Olson, G. M., "From Laboratories to Collaboratories: A New Organizational Form for Scientific Collaboration," *Psychological Science*, Vol. 8, No. 1, January 1997, pp. 28-35.

Goldin, D., "Tools for the Future," Remarks Delivered at the Annual NASA Employee Conference, January, 1998.

Greenhalgh, C., Benford, S., "Boundaries, Awareness and Interaction in Collaborative Virtual Environments," *Proceedings of the 6th International WET-ICE*, June 1997.

Greenhalgh, C., Benford, S., "MASSIVE: A Collaborative Virtual System for Tele-conferencing," *ACM Transactions of Computer-Human Interfaces*, Vol. 2. No. 3, Sept. 1995, pp. 239-261.

Grinter, R.E., "From Workplace to Development: What Have We Learned So Far and Where Do We Go?", *Group '97: Proceedings of the International ACM SIGGROUP Conference on Computer-Supported Cooperative Work*, 1997, pp. 231-240.

Isaacs, E.A., Tang, J.C., Morris, T., “Piazza: A Desktop Environment Supporting Impromptu and Planned Interactions,” *Computer Supported Cooperative Work ’96*, 19996, pp. 315-324.

Johnson, A., Leigh, J., DeFanti, T., Brown, M., Sandin, D., “CAVERN: the CAVE Research Network,” *Proceedings of 1st International Symposium on Multimedia Virtual Laboratory*. March 1998, pp. 15-27.

Jones, K.C., Cygnus, M.W., Storch, R.L., Farnsworth, K.D., "A Virtual Reality for Manufacturing Simulation," *Proceedings of the 1993 Winter Simulation Conference,* 1993, pp. 882-887.

Kriz, R.D., Farkas, D., Batra, R.C., “ Integrating Simulation Research Into Curriculum Modules on Mechanical Behavior of Materials: From the Atomistic to the Continuum," Presented at the MRS 1998 Workshop, Submitted for review to the Journal of Materials Education, January 1999.

Leigh, J., Park, K., Kenyon, R., Wong, H., “Preliminary STAR TAP Tele-immersion Experiments between Chicago and Singapore,” *Proceedings HPCAsia’98*, 1998.

Leigh, J., DeFanti, T., “CAVERN: A Distributed Architecture for Supporting Scaleable Persistence and Interoperability in Collaborative Virtual Environments,” *Journal of Virtual Reality Research, Development and Applications, Vol. 2.2,* December 1997 (1996), pp. 217-237.

Leigh, J., DeFanti, T., Johnson, A., Brown, M., Sandin, D., “Global Tele-immersion: Better Than Being There,” *Proceedings of ICAT '97,* Dec, 1997.

Leigh, J., Johnson, A., DeFanti, T., “CALVIN: an Immersimedia Design Environment Utilizing Heterogeneous Perspectives,” *Proceedings of the IEEE International Conference on Multimedia Computing and Systems'96*, 1996.

Middleton, W.E., McIntyre, III, R.T., O'Keefe, IV, J., "Virtual Reality and Analytical Simulation of the Soldier," *Proceedings of the 1993 Winter Simulation Conference*, 1993, pp. 1048-1052.

Pedersen, E. R., McCall, K., Moran, T. P., Halsz, F.G., “Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings,” *INTERCHI ’93,* pp. 391-398.

Roseman, M., Greenberg, S., “Building Real-Time Groupware with GroupKit, A Groupware Toolkit,” *ACM Transactions of Computer-Human Interaction*, Vol. 3, No. 1, March, 1996, pp. 66-106.

Roussos, M., Johnson, A., Leigh, J., “The NICE Project: Narrative, Immersive, Constructionist/ Collaborative, Environments for Learning in Virtual Reality,” *Proceedings of the ED-MEDIA/ED-TELECOM*, 1997a, pp. 917-922.

Roussos, M., Johnson, A., Leigh, J., Vasilakas, C., Barnes, C., Moher, T., “NICE: Combining Constructionism, Narrative, and Collaboration in a Virtual Learning Environment,” *Computer Graphics*, Vol. 31, Num. 3, ACM SIGRAPH, 1997b, pp. 62-63.

Roussos, M., Johnson, A., Moher, T., Leigh, J., Vasilakas, C., Barnes, C., “Learning and Building Together in an Immersive Virtual World,” *Presence*, 1997c.

Sagar, M.A., Bullivant, D., Mallinson, G.D., Hunter, P.J., "A Virtual Environment and Model of the Eye for Surgical Simulation,", *SIGRAPH '94: Proceedings of the 21st Annual Conference on Computer Graphics*, 1994, pp. 205-212.

Scharm, H., Breining, R. "How the Automotive Industry Uses Immersive Projection Technology," *Proceedings of the 3rd Annual Immersive Projection Technologies Workshop*, 1999, pp. 133-144.

Schiefele, J., Dörr, K., Olbert, M., Kubbat, W., "Stereoscopic Projection Screens and Virtual Cockpit Simulation for Pilot Training," *Proceedings of the 3rd Annual Immersive Projection Technologies Workshop*, 1999, pp. 211-222.

Snowdon, D., Jää-Aro, K., “A Subjective Virtual Environment for Collaborative Information Visualization,” *Virtual Reality Universe'97*, April 2-5, 1997.

Watsen, K., Darken, R.P., Capps, M.V., "A Handheld Computer as an Interaction Device to a Virtual Environment," *Proceedings of the 3rd Annual Immersive Projection Technologies Workshop*, 1999, pp. 51-57.

Wheless, G.H., Lascara, C.M., Leigh, J., Kapoor, A, Johnson, A.E., DeFanti, T.A., “CAVE6D: A Tool for Collaborative Immersive Visualization of Environmental Data,” submitted to *IEEE Visualization 1998*.

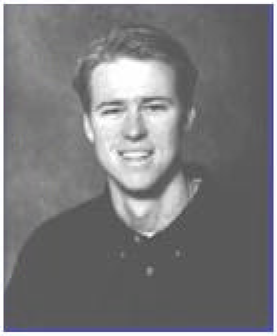
Williams, G., Faste, H., McDowall, I., Bolas, M. "Physical Presence - Palettes in Virtual Space," *Proceedings of the 3rd Annual Immersive Projection Technologies Workshop*, 1999, pp. 65-75.

CITED URLs

Last Referenced: June 24 1999

1. http://www.ncsa.uiuc.edu/VR/cavernus/users.html
2. http://www.evl.uic.edu/cavern/users/nrl.html
3. http://www.evl.uic.edu/cavern/users/vrtime.html
4. http://www.fhw.gr/fhw/en/projects/3d.html
5. http://www.evl.uic.edu/cavern/cavernsoft/
6. http://www.evl.uic.edu/cavern/cavernsoft/limbo.html
7. http://ise.larc.nasa.gov/ISE_Overview/sld051.htm
8. http://scv.bu.edu/HiPArt/intro.html
9. http://www.ice.eecs.uic.edu/~nice/
10. http://www.ncsa.uiuc.edu/VR/cavernus
11. http://bleen.sv.vt.edu/~kcurry/voice.html

Kevin M. Curry



Education

M.S., Computer Science and Applications

VPI & SU, June 24 1999

Thesis Topic: Supporting Collaboration Awareness in Tele-immersion

B.A., History, VPI & SU 1992

Experience

Graduate Research Assistant, Virginia Tech CAVE

(7/98 – 6/99)

Research and development of collaborative virtual environments and collaborative design tools and interfaces for CAVE technology. Full participation at all stages of development; concept and design, implementation, maintenance.CAVE collaborative applications are written using LIMBO and CAVERNsoft APIs to Iris Performer.

Graduate Research Assistant, Dept of Art & Art History

(5/98 - 12/98)

Updating Authorware v. 2.0 tutorial modules to v. 4.0. Design and implementation of new tutorials, including interactive quizzes and searchable indices and glossaries. Tutorial modules are web deliverable through Netscape browsers (v. 3.*, 4.0-4.05) and are used for distance learning in conjunction with a survey course in art history. Digital image correction for online archive database using Adobe Photoshop.

Graduate Research Assistant, Dept. of Mathematics, Cyber1114

(1/97 - 8/98)

Design and implementation of online tutorials to support the teaching of matrix and vector algebras. Tutorials were created in Authorware (v. 3.5, 4.0) and are web deliverable through Netscape browsers. They include animated, interactive demos, and are linked to an electronic text and practice quizzes. Also on a team that created a database of online quiz and exam questions, using WebTeX and WhizQuiz, for use by future instructor

http://www.math.vt.edu/Cyber1114/hometree

Textbook Editor, Department of Mathematics

(5/97 - 7/97)

Revised and rewrote chapters for a solutions manual entitled A Brief Introduction to Matrices and Vectors, using LaTeX and OzTeX

Current Research Interests

Collaborative design tools and interfaces for CAVE

Computer-Supported Cooperative Work

HCI for Computer Assisted Learning and Job Training

Related Coursework

Models and Theories of Human-Computer Interaction (cs 5724)

Computer-Supported Cooperative Work (cs 5734)

Advanced Topics in HCI, Usability Methods and Tool Research (cs 6724)

Systems Simulation (cs 5224)

Information Storage and Retreival (cs 5604)

Languages and Applications

C++/C, Iris Performer, HTML, OpenGL, TeX/LaTeX, JavaScript, Java, Assembly

Lex, YACC, Authorware, MSDev, Cosmo Worlds

Operating Systems

IRIX, Windows NT/95, UNIX , Mac OS

Hardware

SGI O2/Onyx/Octane, PC, Mac, DEC Alpha, DEC Station

Publications

Leigh J., Johnson, A.E., DeFanti, T.A., Brown, M., A Review of Tele-Immersive Applications in the CAVE Research Network, *IEEE Virtual Reality '99*, March, 1999. (Contributing Editor)

Curry, K., Supporting Collaborative Awareness in Tele-immersion, *Proceedings of the IPTW '99*, May 1999.

Activities

Member, Association for Computing Machinery (VT), 1998 - 1999

Member, VT-CAVE Student-Led Users' Group, 1998 -

Co-Chair, VT-CAVE SLUG Tele-immersion Group, 1998 - 1999

Pro-Bono

I typically set aside time to work as a volunteer in the local community.

Some groups that I have worked with:

The Service Learning Center - Harding Elementary 4th grade - Their project was to build interactive history reports using HyperStudio

Blacksburg Presbyterian 3rd and 4th grade Sunday School

If you need help with your project: email kcurry@vt.edu or visit the The Service Learning Center website at Virginia Tech