A Taxonomy of Usability Characteristics in Virtual Environments

Joseph L. Gabbard

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

APPROVED:

Dr. Deborah Hix, Chairperson

Dr. H. Rex Hartson, Member

Dr. Sallie M. Henry, Member

December 3, 1997

Blacksburg, Virginia

Keywords: Virtual Reality, Virtual Environments, Usability, User Interface Process Improvement

A Taxonomy of Usability Characteristics in Virtual Environments

Joseph L. Gabbard

(ABSTRACT)

Despite intense and wide-spread research in both virtual environments (VEs) and usability, the exciting new technology of VEs has not yet been closely coupled with the important characteristic of usability — a necessary coupling if VEs are to reach their full potential. Although numerous methods exist for usability evaluation of interactive computer applications, these methods have well-known limitations, especially for evaluating VEs. Thus, there is a great need to develop usability evaluation methods and criteria *specifically* for VEs. Our goal is to increase awareness of the need for usability engineering of VEs and to lay a scientific foundation for developing high-impact methods for usability engineering of VEs.

The first step in our multi-year research plan has been accomplished, yielding a comprehensive multi-dimensional taxonomy of usability characteristics specifically for VEs. This taxonomy was developed by collecting and synthesizing information from literature, conferences, World Wide Web (WWW) searches, investigative research visits to top VE facilities, and interviews of VE researchers and developers.

The taxonomy consists of four main areas of usability issues: *Users and User Tasks in VEs*, *The Virtual Model*, *VE User Interface Input Mechanisms*, and *VE User Interface Presentation Components* Each of these issues is progressively disclosed and presented at various levels of detail, including specific usability suggestions and context-driven discussion that include a number of references. The taxonomy is a thorough classification, enumeration, and discussion of usability issues in VEs that can be used by VE researchers and developers for usability assessment or simply design.

This research (award number: N00014-96-1-0385) is funded by the Office of Naval Research under Dr. Helen Gigley, Program Manager.

Dedication

I would like to dedicate this thesis to all of those who supported me, encouraged me, and believed in me during my pursuit of this knowledge. To my love Christy for her patience and understanding, ideas and opinions, undying love, and desktop delivery of warm and wonderful meals. To my parents for their love, enthusiasm for the college atmosphere, for raising me to believe that there is nothing I can't do, and for never questioning my goals and ambitions. To my loving sister, Jennifer, for reminding me never to worry about what "other people" think. To Dr. Deborah S. Hix for her patience, relentless grammar and spell checking, and most of all, her understanding that there is more to life than graduate research. To Dr. Sallie Henry, Dr. Jules Lindau, and Dr. Kevin Wilkes for convincing me that graduate school research was inevitably part of my life's path.

In Virginia Tech's Computer Science office, I would like to thank Tammi, Jessie, and Jo-anne for their unquestioning willingness to help even the lowliest of graduate students at the drop of a hat. To Neill Kipp and Matthew McMahon for the weekly El Rodeo lunches where stimulating conversation about life and love was "el especial del dia." To the members of Eetwau, who always understood that work came before play...ing, "out" that is. Likewise, to Steve Callahagn for knowing when to stay away, and when to offer study breaks.

And finally to the creatures; my cat Cassidy (a.k.a. "Newt", "New-er", "Nassidy", and "Nass") for the many years of companionship and for always finding a way on top of my research (something I always had a hard time doing!), and to "the puppies", Sable and Tramp for their company and protection — ensuring that, even when I was home alone, I was not home alone.

Acknowledgments

This research (award number: N00014-96-1-0385) is funded by the Office of Naval Research under Dr. Helen Gigley, Program Manager. We would like to thank Dr. Gigley for her continued support of an on-going synergistic collaboration in human-computer interaction research between Virginia Tech and NRL over the past several years.

We would also like to thank the reviewers of this taxonomy for their time, effort, and suggestions. In particular, we would like to thank:

- Dr. W. Barfield, Virginia Tech,
- D. Bowman, Georgia Tech,
- Dr. R. Darken, Naval Postgraduate School,
- J. Goldman, SRI,
- Dr. E. Swan, NRL, and
- Dr. J. Templeman, NRL

We are very lucky to have had the opportunity to have our work reviewed by these outstanding professionals in the field. This type of peer review can only improve the quality of both research and publication.

We would also like to thank all of those who graciously entertained us during our investigative research visits. The information gleaned from these visits is the type of information not available through any other means, especially the hands-on experience in some of the field's most notable research facilities and programs. As such, interaction with the following individuals was integral in development of the taxonomy;

- Dr. Randy Pausch and Dr. Matthew Conway — Univ. of Virginia, Charlottesville VA
- William Sherman — NCSA, Univ. of Illinois at Urbana-Champaign, Urbana-Champaign IL
- Pat Beard — EDS Virtual Reality Center, Detroit MI

- Dr. Glen Osga and Elaine Schiller — NRAD, San Diego CA
- Frank Hepburn — Kaiser Electro-Optics, Carlsbad CA
- Dr. Rudy Darken and Bill Cockyne — Naval Postgraduate School, Monterey CA
- Dr. James Templeman, Linda Sibert, Dr. Lawrence Rosenblum, Jim Durbin, Dr. J. Edward Swan, Brad Colbert, and Chris Scannell — NRL, Washington DC
- Martin McGurn — High Techsplanations Inc., Rockville MD
- Dr. John Sibert — George Washington University, Washington DC
- Eric Lorimer — Fakespace, Menlo Park CA
- Randy Stiles – Lockheed Martin, Palo Alto CA
- Jon Goldman — SRI, Menlo Park CA
- Suzanne Weghorst — Univ. of Washington, Seattle WA
- David W. Mizell — Boeing, Seattle WA

We thank them not only for their time and effort, but for their insights, suggestions, and hospitality.

We would also like to thank our colleagues in Blacksburg for their help on the taxonomy and, more importantly, for their continued commitment to Virginia Tech's HCI and VE programs. These people include Dr. Rex Hartson, Dr. Sallie Henry, Dr. Bob Williges, Dr. Woodrow Barfield, John Kelso, Dr. Mike Snow, Dennis Neale, Mike McGhee, and Brian Amento.

Contents

List of Figures

List of Tables

1 Introduction

"Techniques and tools for interacting with virtual environments are at the core of research and development efforts around the world." Thus begins the Introduction to a Special Issue of the *ACM Transactions on Computer-Human Interaction* on Virtual Reality Software and Technology [Singh and Feiner, 1995]. Military, government, commercial, and industrial organizations are investing enormous amounts of effort and resources to produce virtual environments (VEs). While VEs have been gaining broad attention, usability of the user interface has become a major focus of interactive system development. Yet despite intense and wide-spread research and development in both VEs and usability, the exciting new technology of VEs has not yet been closely coupled with the important characteristic of usability — a necessary coupling if VEs are to reach their full potential.

1.1 Objective

The objective of the research reported in this thesis was to develop, in detail, a multi-dimensional taxonomy of usability characteristics of VEs. This taxonomy, described in detail in Section 3, will serve as a foundation for evolving *high-impact* usability methods for development of VEs, based on a scientific structured — as opposed to the typical ad hoc — approach.

The taxonomy can be used in a variety of ways. For example, current usability evaluation methods can be assessed to determine which of the taxonomy characteristics are addressed and which are not. As such, this taxonomy serves as a framework for analysis, discussion, comparison, definition, research, development, and evaluation of usability evaluation methods. In this phase of our research we are not looking for new evaluation methods per se, but rather for a foundation upon which development of new usability evaluation methods for VEs can be based in future work that will grow seamlessly from this work.

1.2 Motivation and Problem Statement

The goal of much work in VEs thus far has been to produce "gee whiz" technology; until recently, there has been very little usability-focused research in VEs. More likely, unsubstantiated claims of improved performance and user satisfaction are based at best on a few user interface guidelines and at worst on warm fuzzy feelings of VE developers and "way cool" comments from VE users. The few reported evaluations are rarely comprehensive. For example, novel interaction techniques developed by van Dam et al. at Brown University are routinely evaluated for usability, but seldom are the techniques set in a realistic application and used to perform "real" user tasks within that application. An underlying assumption among both researchers and developers sometimes seems to be that VEs, because they are a novel and impressive technology, are inherently good and usable. Progress is needed to move beyond this flawed assumption, to have usability engineering become a routine activity in VE development, with methods to produce VEs that are effective and efficient for their users, not merely new and different.

Most VE literature to date applies either to a specific VE application (with specific instances of goals, interaction techniques, equipment, etc.) or to a common group of usability concerns (head-mounted displays (HMDs), haptics, collaboration, etc.), with little or no explicit usability references. In some cases, one can infer usability information from specific findings by "reading between the lines". However, high usability is not something that happens by accident or by good luck; through specific usability methods (e.g., [Hix and Hartson, 1993] [Nielsen, 1993]), it is engineered into a product from the beginning and throughout the development life cycle.

Nonetheless, there is beginning to be at least some awareness of the need for usability engineering within the VE community. There are a handful of articles which address usability concerns for particular parts of the VE usability space. For example, some have published guidelines for spatial input devices (e.g., [Hinckley et al., 1994a]), hints for three dimensional interfaces design (e.g., [Bricken, 1990]), and usability issues in haptic feedback hardware (e.g., [Hannaford and Venema, 1995]). However, many publications that include usability issues fail to address the complex inter-dependencies present in VEs among users, tasks, input devices, output devices, etc.

1.3 Limitations of Existing Usability Methods

Although numerous methods exist for usability engineering of interactive computer applications, these methods have well-known limitations, especially for VEs. For example, many existing usability engineering methods are time consuming and personnel intensive. Most methods are applicable only to a narrow range of user interface types (e.g., GUIs — graphical user interfaces) and have had little or no use in developing innovative, non-routine interfaces such as those found in VEs. Further, VEs have interaction styles so radically different from ordinary user interfaces that traditional user-task-based development and evaluation methods may be neither appropriate nor effective for all situations.

The focus of most existing methods, while properly user-task-based, is on a single user performing isolated, low-level user tasks — very different than the typical VE in which one or more users are performing integrated, shared, multi-threaded tasks. But a focus "in the large" is particularly important for more complex tasks and more unusual interfaces such as those inherent in VEs. Many characteristics that are unique to VEs and are key to their usability, such as perceived presence and perceived real world fidelity, are not addressed in existing usability methods; they do not support, for example, quantification (or even qualification) of a user's perception of such characteristics. Traditional user-task-based methods do not consider VEs in which two or more users interact; this requires development for multiple users using different sets of hardware, perhaps even at different physical sites. Identifying such differences between well-established GUIs and novel new VEs exposes the shortcomings of applying traditional GUI-related usability methods to VEs, and hence strongly motivates the need for research in development of usability engineering methods specifically for VEs.

"One of the components of virtual reality is the insertion of humans into the virtual environments (that are being created)" [Darken, 1996]. Successfully inserting humans into

VEs implies that user interfaces of such VEs be able to support realistic work and tasks intuitively. An understanding and assessment of human performance and satisfaction lie at the heart of developing, evaluating, and improving usability of VEs.

Thus existing methodologies need extensive assessment and modifications to support invention, development, and study of VE user interfaces. Further, there is a need to produce a new generation of high-impact methods for usability engineering of VEs. By *high-impact*, we mean *effective*, *low cost*, *fast*, and *easy to use*, and, in this case, *applying specifically to VEs.* This is a time-critical topic; as numerous VEs are being built without any focus on usability, it becomes increasingly important to find approaches and methodologies for remedying this situation. The field of VEs is now mature enough for research in user-centered development methods — a largely unexplored topic — to be not only fruitful, but critical, to the optimal, cost-effective production, evaluation, and use of this promising technology.

But challenges to producing usability engineering methods for VEs include lack of a framework as a structured basis for method development. An in-depth literature review in a recent Virginia Tech Ph.D. dissertation [Snow, 1996] confirms a scarcity of foundational work for VE development, especially for usability engineering methods. The utility of taxonomies has been stressed, for example, by [Neale and Carroll, 1997], which states that "one of the first and most important tools for developing and evaluating (new) metaphors are comprehensive taxonomies." As a major step in creating new methods for usability engineering of VEs, we have produced a comprehensive *multi-dimensional taxonomy of usability characteristics specifically for VEs.*

1.4 Scientific Significance of This Taxonomy

This taxonomy can have both immediate and long-term impact on the field of VEs. In the short term, it comprehensively defines a multi-dimensional structure of characteristics important for usability in VEs, and can support classification and evaluation of existing usability engineering methods, pinpointing their strengths and weaknesses for VE development and evaluation. The design space for VEs is far greater than that for traditional user interfaces such as GUIs. Thus, there is a need for techniques to assist in systematic reduction of the space, given a particular VE application goal. This taxonomy does not trivialize the inherently difficult job of designing usable VEs, but it does provide guidance, structure, and focus for it. The taxonomy accumulates a large "memory", or body of knowledge about VEs that includes their most important usability characteristics, providing an unbiased acquisition and organization of relevant information.

In the longer term, the taxonomy will, perhaps more importantly, provide a basic, scientific foundation for evolving a new generation of high-impact methods for usability engineering of VEs. These new methods will come both from modification of existing methods so they accommodate VEs, as well as from altogether new approaches to usability engineering of VEs.

This taxonomy is useful for VE researchers and developers, as well as funding agencies. Specifically, researchers and developers can get a breadth and depth overview of usability characteristics that are important to VEs, and can find guidance, via extensive literature references and annotated discussions, for answering design questions for VE applications they are producing. Funding agencies, as well as researchers, will find a variety of open research questions that we have explicitly listed in the discussion section of the taxonomy, as well as the almost endless roster of open research questions that are implied by "gaps" in the taxonomy. Further, it can guide the rational design of costly empirical studies, for example, to compare various kinds of VE devices, by providing current information and pinpointing known contradictions in existing knowledge. Presently the limited empirical studies about VEs are performed based largely on what is tenable, not necessarily on what is most critical or most basic. This taxonomy can help focus the almost unlimited issues into those that will provide the most important results most cost-effectively.

1.5 Strengths and Weaknesses of the Taxonomy

One of our goals for this taxonomy is to create an easy-to-use reference for VE developers. As such, it is imperative that thesis organization provide non-linear access to thesis content in a meaningful, relevant fashion. The way in which we present this taxonomy is only one of many possible organizations. Although we based the structure on Norman's theory of action (see Section 3), many readers may find that the basic structure is obvious, arbitrary, or even simplistic. The strength here is not so much in producing a perhaps evident structure for usability characteristics of VEs, but rather in the breadth of the characteristics and especially in the in-depth literature research and review that accompany it. A benefit of this structure is that it is easily extensible as new interaction techniques, devices, hardware, and other usability issues for VEs emerge.

We do not claim this taxonomy to be exhaustive, nor do we claim the discussions that accompany it and cite relevant literature are exhaustive (but perhaps they are exhausting!). While this is an attempt to thoroughly enumerate usability issues in VEs, the taxonomy is in its first incarnation and, as such, parts of it may have serious omissions, or may even be erroneous. While it points out, when possible, recognized contradictions in prevailing knowledge, future work in the fast-moving field of VEs may render elements of the current taxonomy incorrect. However, a strength is the extensive collation and comparison of relevant work, within the taxonomic structure. Our comprehensive presentation serves as both an aggregate and a filter for VE usability-related information, so that a reader can find areas of general interest, look at the comparative discussion, and then go to citations directly for those situations in which more complete details are desired. So the taxonomy itself is a stand-alone organization of usability characteristics of VEs, but the accompanying context-driven discussion is not necessarily stand-alone.

This taxonomy is particularly important to instill in VE researchers, developers, and even users the importance of user-centered approaches to VE development. In fact, many VE researchers and developers commonly ask about designing VEs, "Why are we looking for new metaphors and techniques; why don't we just do it in the VE the way people are used to doing it in the real world?" Some results that are reported herein give support to the idea of deviating, possibly even dramatically, from real world analogy. Innovation may lead to even better ways of performing tasks in a VE than we can perform similar tasks in the real world. The key is constant, thorough user-based development and evaluation to ensure that, in fact, human performance and satisfaction are increased by new VE interaction techniques and applications.

A deficiency of this taxonomy is that it does not explicitly report in detail *how* to implement a given usability suggestion. Pointers to outside references may serve this purpose best, since a number of articles on a particular topic can yield more detailed information than can a single document. Some of the references cite conclusions regarding VE design that could equally apply to GUIs, not just to VEs. In those cases, we have attempted to state those that appear most important, to give them increased visibility because of their potential impact on VE usability.

A further deficiency is that the underlying framework has not yet been extensively shown to be useful for developing classifications within the design space for which it is intended — a key hallmark of a taxonomy [Fleishman and Quaintance, 1984]. This taxonomy needs further use and extension to include well-defined rules for classifying specific VEs, as well as more formal evaluation to show the efficacy of such rules. Once such rules exist, the act of classifying a particular VE should immediately reveal information about characteristics in terms of expected human performance, merely by the relative association and comparison of similar systems [Darken, 1997].

2 Research Approach and Methods

2.1 Developing the Taxonomy

Our approach to developing a taxonomy of usability characteristics was to collect and synthesize information from many different sources, so that the taxonomy is comprised of a structured collection of otherwise piecemeal findings derived from research and experience. Information for the taxonomy was derived from several different "sources of inspiration" including:

- VE-related journals and publications; of particular note are
  Presence: The Journal for Telepresence and Teleoperation,
  the yearly *Virtual Reality Annual International Symposium(VRAIS)* proceedings,
  and the annual *Human Factors in Computing Systems Conference (CHI)* proceedings;
- Investigative research visits to some of the top VE facilities in academia, industry, and government (a detailed list of visits is given in Appendix A);
- *SIGGRAPH '96 Conference* notes, panels, exhibits, discussions, and observations;
- Human-computer interaction (HCI) related literature;
- Structured interviews administered during the investigative research visits; and
- World Wide Web searches for VE-related work.

These "sources of inspiration" were investigated more or less in parallel. Of note were specific usability findings from articles, novel interaction techniques and design suggestions in the literature, obvious usability problems observed during investigative visits, and comments made by those interviewed as well as *SIGGRAPH '96* speakers.

What began as an ever-growing "laundry list" of characteristics and observations was iteratively structured into a taxonomy by identifying and grouping characteristics with common usability implications. Because such a taxonomy can potentially address **every** aspect of VE interaction and technology, defining a single unified hierarchical model is not a trivial task. As such, we do not make any claims about the completeness of this taxonomy, as mentioned in Section 1.5. We have, however, included every aspect we encountered in our "sources of inspiration," and therefore consider the taxonomy to be comprehensive.

The taxonomy was evaluated via qualitative, anecdotal, and informal analysis. Drafts outlining thesis structure and content, as well as a draft of the nearly completed thesis, were iteratively critiqued by several researchers in the field of VEs and usability engineering (W. Barfield, Virginia Tech; D. Bowman, Georgia Tech; R. Darken, Naval Postgraduate School (NPS); J. Goldman, SRI; E. Swan, Naval Research Lab (NRL); J. Templeman, NRL). The comments and insight they provided were very useful in determining the quality of the taxonomy (e.g., correctness of classification, relevance of interrelationships, applicability of discussion, etc.), and in making modifications and extensions to it.

The taxonomy was approximately 90% complete when sent to review. At this point, the taxonomy's structure was fully developed, however, taxonomy content was not yet complete. Four of the six reviewers provided comments in a timely fashion, providing comments on both the structure and content. Examples of editor suggestions include: "provide a labeling scheme for the specific usability suggestions", "facilitate access from context driven discussion back into tables", and "point out the few most well-known instances where the literature contradicts itself." The suggestions were ranked according to impact on taxonomy and time to complete. Not all suggestions were implemented. For the complete list of editor's comments and suggestions see Appendix B. Ultimately, evaluation and acceptance of the taxonomy will come from VE developers and researchers as case-specific VEs are analyzed according to taxonomy categories.

2.2 Thesis Organization

The taxonomy is first presented in a condensed, hierarchical structure; a diagram depicting high-level relationships among four areas of usability issues. Each group of usability issues is then expanded into several tables summarizing usability findings and specific suggestions, with appropriate section and citation references. Moreover, the tables serve as a thesis and reference "map" into taxonomy discussion (see Section 4 for more details). In fact, the bulk of this thesis is geared toward taxonomy discussion, which presents the four areas of usability issues in the following sections:

- Users and User Tasks in VEs (see Section 5);
- The Virtual Model (see Section 6);
- VE User Interface Input Mechanisms (see Section 7); and
- VE User Interface Presentation Components (see Section 8).

In each of these sections, a general discussion of usability characteristics is presented, followed by a more detailed "context-driven" discussion where relevant characteristics are addressed in terms of specific tasks, interaction techniques, hardware, etc. Section 5 (Users and User Tasks in VEs) examines general task characteristics and types of tasks in VEs. Section 6 (The Virtual Model) discusses usability characteristics of generic components typically found in VEs. Section 7 (VE User Interface Input Mechanisms) identifies characteristics of VE input devices. Section 8 (VE User Interface Presentation Components) examines usability characteristics of VE output devices. Section 9 proposes some future research opportunities that we hope to explore in the coming months.

Throughout the thesis, we have included some open-ended research questions that surfaced during taxonomy research efforts. They are presented in the margins just outside related discussion.

3 Overview of Taxonomy

We have organized the taxonomy to facilitate non-linear access to information both within and outside this thesis. In particular, we have devised a "top-down" approach to presenting taxonomy issues and discussion that include:

- Overview of Taxonomy Areas (diagram);
- Specific Usability Suggestions and Considerations (tables);
- Context-Driven Discussion; and
- References (for all cited works, including WWW addresses where appropriate and available.

At the highest level, the taxonomy is presented in an abstract hierarchical structure represented by the four shaded boxes and their connections shown in Figure 1. This diagram depicts high-level relationships among the taxonomy's four major areas of usability issues, mentioned in Section 2.2. The diagram also contains another level of refinement for each of these major areas, shown as white boxes in Figure 1. For example, "VE User Interface Presentation Components" is refined into "Visual Feedback", "Haptic Presentation", "Aural Feedback", and "Environmental Feedback and Other Presentations". For every box within the diagram there is a corresponding table, which, in turn, presents specific usability suggestions and considerations. Arrows in the diagram represent "information" flow between user and VE. That is, user actions through input devices alter the state of the virtual model which is then presented to the user via presentation components.

Structuring high-level taxonomy areas and usability characteristics within each area was one of our biggest challenges. Indeed, the space of usability characteristics in VEs does not fit into a single "natural" or "correct" organization or ordering. However, some ordering had to be imposed on the characteristics, revealing and restricting relationships as dictated by that particular structure. One approach to ordering a space of characteristics is to use general theories of human-computer interaction as a guide. After reviewing several theories and models, we found Norman's *theory of action* [Norman, 1990] to be an

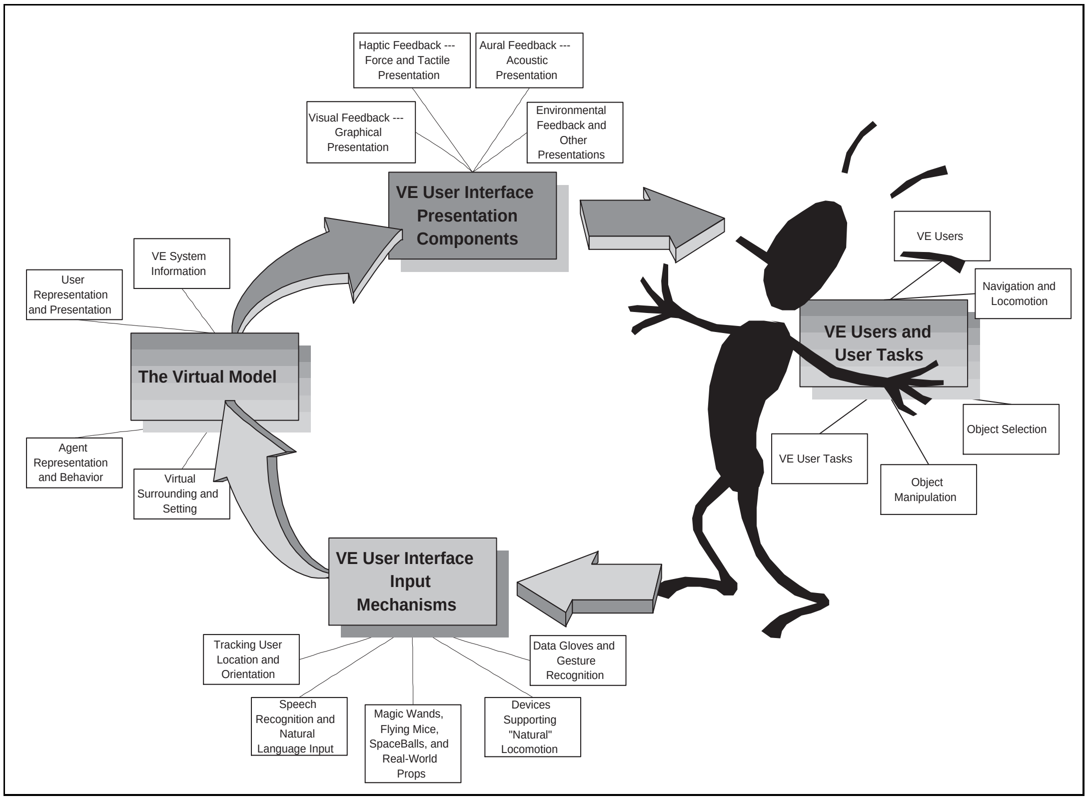

Figure 1: An Overview of Taxonomy Areas

appropriate foundation upon which to base our current organization. This theory of action defines several stages of activity and associated interdependencies that are inherent in any interaction between human and machine [Norman, 1990]. It consists of several stages of user activities involved in a user's performance of a task. We found this framework to be particularly well-suited for addressing how individual usability issues fit into a more abstract, larger scale understanding of interaction between users and VEs.

In particular, Norman defines a "gulf of execution", which is bridged when the *commands and mechanisms* of an interactive system (in our case, VEs) match the thoughts, goals, and intentions of a user. This we mapped into our taxonomy as "VE User Interface Input Mechanisms". Norman also defines a "gulf of evaluation", which is bridged when *output* presents an appropriate conceptual system model that a user can readily perceive, evaluate, and understand. This we mapped into our taxonomy as "VE User Interface Presentation Components". Further, the theory of action contains the physical *system*, which we abstracted into the "Virtual Model" and also contains *goals* of the user, which we abstracted into "VE Users and User Tasks". Thus, the four major areas shown in Figure 1 are strongly influenced by corresponding components in the theory of action, and the flow is strongly influenced by the theory's corresponding flow.

3.1 Specific Usability Suggestions and Considerations

The first level of refinement for each major area of usability issues is expanded into tables summarizing usability findings and specific suggestions. These tables (found in Section 4) also contain pointers to supporting sections in the context-driven discussion, as well as citations in the reference list. Thus, these tables serve as a thesis and reference map into lower levels comprised of sections containing detailed discussion of each taxonomy area. There is one table corresponding to each white box in Figure 1.

A labeling scheme has been developed to easily distinguish between and reference specific usability suggestions. This scheme assigns a unique *suggestion label* to each usability suggestion. The labeling scheme is helpful while reading the context-driven discussion as each particular usability suggestion can be easily identified by its enclosure within $\ll$ $\gg$, which references back to its corresponding suggestion label.

3.2 Context-driven Discussion

Each of the four discussion sections — one for each major area of the taxonomy, listed above — begins with a general presentation of usability characteristics specific to that area. This is followed by an in-depth context-driven discussion for the lower levels of refinement in the taxonomy, in which relevant usability characteristics are addressed in terms of specific tasks, interaction techniques, hardware, etc. Issues are compared and contrasted, and apparent contradictions in research findings are elaborated. These discussions comprise the bulk of this thesis.

We have provided additional visual and navigational cues to explicitly connect context-driven discussion back to specific usability suggestions (contained in specific usability suggestions tables discussed above). Within context-driven discussion, characteristics, suggestions, considerations, and issues relevant to VE usability are printed in ***bold italics***, and contain backward pointers , indicated by ≪ ≫, to relevant numbered usability suggestions in these tables.

3.3 Use of the Taxonomy

As mentioned in Section 1.4, our long-term goal for use of the taxonomy is for it to serve as a foundation for development of new methods for usability engineering specifically for VEs. However, in the short term, it will serve as the "memory", also mentioned in Section 1.4, for collecting a comprehensive set of literature about VE usability issues. This collection can then be used, for example, by a VE developer who wishes to learn more about usability issues related to navigation in VEs.

Starting at the top-most level in the diagram (Figure 1), the developer sees that one of the boxes under "VE Users and User Tasks" is labeled "Navigation and Locomotion". From here, the developer can read through the table labeled "Navigation and Locomotion" to get an idea of the types of usability concerns inherent in VE navigation. From the table, the developer can either access taxonomy discussion on navigation (in the context-driven discussion section entitled "Navigation and Locomotion") or get an overview of published literature and citations in the field of VE navigation by noting the references cited within the table and/or discussion. For example, if a developer is looking for information about alternative ways of navigating in VEs, the developer can look at the discussion section entitled "Navigation and Locomotion Metaphors" to find a discussion and references about three possibilities: Camera-(or eyeball-)in-hand, Scene-in-hand, and Flying. We are currently making this process of successive refinement more "usable" by hyper-linking diagram, tables, discussion, and references in the Web-based version of the taxonomy, as mentioned in Section 9.

As another simple example of taxonomy use, someone wishing to learn more about VE interaction devices in general could follow Figure 1 into the tables for "VE User Interface Input Mechanisms", and then read more details about many different kinds of input devices, including tracking user location and orientation, devices dupporting "natural" locomotion, data gloves and gesture recognition, magic wands, flying mice, spaceballs, and real-world props, and speech recognition and natural language input in the related context-driven discussion section.

4 Specific Usability Suggestions

Presented below are the tables containing specific usability suggestions and considerations. Note that although the suggestions in each table are presented in an active tone, none of the suggestions should be taken or followed out of context. That is, the suggestions given in the tables are powerful, and most likely apply to particular arrangements of VE users, tasks, hardware, applications, etc. *Blindly applying the suggestions will not make a VE instantly usable.* The purpose of the taxonomy discussion and reference list, discussed below, is to give the necessary context in which to assess and appropriately apply usability suggestions.

4.1 Users and User Tasks in VEs

TABLE 1 Some Usability Issues of VE Users

VE Users			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Users1	Take into account user experience (i.e., support both expert and novice users)	37	[Egan, 1988]
Users2	Support users with varying degrees of domain knowledge	37	[Egan, 1988]
Users3	Take into account users' technical aptitudes (e.g., *orientation*, *spatial visualization*, and *spatial memory*)	38	[Egan, 1988] [Darken and Sibert, 1995] [Stoakley et al., 1995] [Stanney, 1995]
Users4	Support both right *and* left-handed users (e.g., through devices)	38	
Users5	Accommodate natural, unforced interaction for users of varied age, gender, stature, and size	38	[Kaiser Electro-Optics, 1996] [University of Washington, 1996] [Boeing, 1996]

Table 2 Some Usability Issues of VE User Tasks

VE User Tasks			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Tasks1	Take into account the number and locations of potential users	39	
Tasks2	When designing collaborative VEs, support social interaction among users (e.g., *group communication, role-play, informal interaction*)	39	[Waters et al., 1997]
Tasks3	In collaborative VEs, support cooperative task performance (e.g., *facilitate social organization*, *construction*, and *execution of plans*	40	[Malone and Crowston, 1990] [Benford, 1996]
Tasks4	Provide awareness-based information for competitive task performance	41	
Tasks5	Support concurrent task execution	42	
Tasks6	Design interaction mechanisms and methods to support user performance of *serial* tasks and task sequences	42	
Tasks7	Provides stepwise, subtask refinement including the ability to *undo*	43	

Table 3 Some Usability Issues of Navigation and Locomotion

Navigation and Locomotion			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Nav1	Support appropriate types of user navigation (e.g., *naive search*, *primed search*, *exploration*)	45	[Darken and Sibert, 1996]
Nav2	Facilitate user acquisition of survey knowledge (e.g., *maintain a consistent spatial layout*)	45	[Darken and Sibert, 1995] [Darken and Sibert, 1996] [Lynch, 1960]
Nav3	When designing landscape and terrain layout, consider [Darken and Sibert, 1995] *organizational principles*	45	[Darken and Sibert, 1995] [Darken and Sibert, 1996]
Nav4	When appropriate, include *spatial labels, landmarks, and a horizon*	46	[Darken and Sibert, 1995] [Darken and Sibert, 1996] [Bennett et al., 1996]
Nav5	Provide information so that users can always answer the questions: Where am I now? What is my current attitude and orientation? Where do I want to go? How do I travel there?	46	[Wickens and Baker, 1995]
Nav6	Avoid *mode-based* navigation	48	[Fairchild et al., 1993]
Nav7	Strive for body-centered interaction	48	[Slater et al., 1995b] [Davies, 1996]
Nav8	Choose control metaphor(s) that naturally match the application task space	50	[Neale and Carroll, 1997] [Fairchild et al., 1993]
Nav9	Choose control metaphor(s) that allow for concurrent task execution	42, 50	[Fairchild et al., 1993]
Nav10	Ensure that point to point animations do not restrict *situational awareness*	51	[Wickens and Baker, 1995] [Pausch et al., 1996] [Bowman et al., 1997]
Nav11	Use body-based steering to support concurrent manipulation tasks	52	[Templeman, 1996] [Slater et al., 1995b] [Davies, 1996]
Nav12	Use head-based steering approaches when direction of gaze and travel are logically connected or for simple object-to-object movements	53	[Bowman et al., 1997] [Vivid, 1997]
Nav13	For VEs with little or no manipulation, sitting users, or limited facility space, consider hand-based steering approaches	54	[Templeman, 1996]

Table 4 Some Usability Issues of Object Selection

Object Selection			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Select1	Use direct manipulation for selections based on spatial attributes (e.g., *location, orientation, shape)*	57	[Shneiderman, 1992] [Hutchins and Norman, 1986]
Select2	When selecting distant objects via direct manipulation, exaggerate object size, appearance, inter-object distances, etc.	57	[Gibson, 1986] [Mine et al., 1997]
Select3	Facilitate selection of multiple objects	57	[Templeman, 1997b]
Select4	Use *bounding boxes*, *marquees*, *rubber bands*, etc., for multiple selections based on spatial relationships	57	[Mapes and Moshell, 1995]
Select5	Use non-direct manipulation means (such as *query-based selection*) when selection criteria are temporal, descriptive, or relational	58, 70	
Select6	Supply users with appropriate selection feedback (e.g., *highlighting, outlining*, *acoustic* or *verbal confirmation*)	58	
Select7	Use transparency to avoid occlusion during selection	59	[Hinckley et al., 1994a] [Zhai et al., 1994]
Select8	Strive for high frame rates and low latency to assist users in three-dimensional target acquisition	60	[Ware and Balakrishnan, 1994] [Richard et al., 1996]
Select9	Object selection points should be made as obvious and accessible as possible	60	
Select10	Use damping, snapping, and/or trolling to aid in selection of objects	61	[Hix et al., 1997]
Select11	Targeting of three-dimensional objects should be based upon relative motion	61	[Hinckley et al., 1994a]
Select12	Use ray casting when objects to be selected are very small or co-located among many others	62	[Hinckley et al., 1994a]
Select13	Use *spotlighting* or other visual cues when selecting via cone casting	63	[Liang and Green, 1994]

Table 5 Some Usability Issues of Object Manipulation

Object Manipulation			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Manip1	Provide accurate depiction of location and orientation of surfaces	64	[Wickens and Baker, 1995]
Manip2	Minimize display lag	65	[Wickens and Baker, 1995] [Sturman et al., 1989]
Manip3	Support multimodal interaction	65	[Wickens and Baker, 1995] [Brooks et al., 1990]
Manip4	Provide spatially relevant and revealing user point of view	65	[Wickens and Baker, 1995] [Mine et al., 1997]
Manip5	Avoid non-intuitive, unnatural, or poorly mapped gesturing	65	[Mapes and Moshell, 1995]
Manip6	When using pinch gloves, keep in mind user experience when determining the number of modes or pinching combinations	66	
Manip7	Support two-handed interaction (especially for *manipulation-based* tasks)	38, 66	[Hauptmann, 1989] [Hinckley et al., 1994a] [Guiard, 1987]
Manip8	For two-handed manipulation tasks, assign dominant hand to fine-grained manipulation *relative* to the non-dominant hand	38, 67	[Hinckley et al., 1994a] [Guiard, 1987]
Manip9	When rotating objects through large angles via natural wrist rotation, employ some form of clutching or ratcheting mechanism	68	[Hinckley et al., 1994a]
Manip10	Allow users to alter basic object attributes (e.g., *color*, *shape*, *labels*)	70	
Manip11	When possible, combine query formation with selection methods	58, 70	[Esposito, 1996]
Manip12	Support interface query for users to determine what actions are available for objects	71	[Esposito, 1996]

4.2 The Virtual Model

Table 6 Some Usability Issues of User Representation and Presentation

User Presentation and Represenation			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
UserRep1	For collaborative VEs, design avatars to convey user viewpoint and activity	79	[Benford et al., 1995]
UserRep2	Ensure that users' avatars provide a familiar, accurate, and relavent frame of reference	79	[Benford et al., 1995]
UserRep3	Provide egocentric point of view(s) when users need to experience a strong sense of self-presence	79	
UserRep4	Provide exocentric view(s) when relative positioning and motion between user and objects are important	79	
UserRep5	User embodiments should be as efficient as possible (e.g., useful and relevant *content*, *detail*, and *sensory representation*)	80	[Benford et al., 1995]
UserRep6	Allow users to control presentation of both themselves and others (e.g., support *graceful degradation*)	80	
UserRep7	Allow users to alter point of view, or viewpoint (i.e., support the ability to view scenes and objects *from many different* angles)	81	[Stoakley et al., 1995]

Table 7 Some Usability Issues of VE Agent Representation and Presentation

VE Agent Presentation and Represenation			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Agents1	Include agents that are relevant to user tasks and goals	84	[Trias et al., 1996] [Ishizaki, 1996]
Agents2	Real-world, high-fidelity physical and behavioral agent representation may be useful for training and simulation VEs	84	[Ishizaki, 1996]
Agents3	Allow agent behavior to dynamically adapt, depending upon context, user activity, etc.	84	[Trias et al., 1996]
Agents4	Represent interactions among agents and users (rules of engagement) in a semantically consistent, easily visualizable manner	85	[Trias et al., 1996]
Agents5	Organize multiple agents according to user tasks and goals	86	[Ishizaki, 1996]

Table 8 Some Usability Issues of Virtual Surrounding and Setting

Virtual Surrounding and Setting			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Setting1	Use setting to increase user presence	86	[Barfield et al., 1995]
Setting2	Exploit real-world experience, by mapping desired functionality to everyday items (e.g., *clock to convey time*)	86	[Neale and Carroll, 1997]
Setting3	Use relevant settings that suggest user activity and tasks	87	
Setting4	Employ rendering techniques that support detailed presentation of setting without introducing lag	88	[Oshhima et al., 1996]

TABLE 9 VE System and Application Information

VE System and Application Information			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
SysInfo1	Use progressive disclosure for information-rich interfaces	89	[Hix and Hartson, 1993]
SysInfo2	Pay close attention to the visual, aural, and haptic organization of presentation (e.g., eliminate *unnecessary information*, minimize *overall and local density*, group *related information*, and emphasize information *related to user tasks*)	89	[Hix and Hartson, 1993]
SysInfo3	Strive to maintain interface consistency across applications	90	[Hix and Hartson, 1993]
SysInfo4	Language and labeling for commands should clearly and concisely reflect meaning	90	[Hix and Hartson, 1993]
SysInfo5	System messages should be worded in a clear, constructive manner so as to encourage user engagement (as opposed to user alienation)	91	[Hix and Hartson, 1993]
SysInfo6	For large environments, include a navigational grid and/or a navigational map	91	[Darken and Sibert, 1995]
SysInfo7	When implementing maps, adhere to [Darken and Sibert, 1995] map design principles	92	[Darken and Sibert, 1995]
SysInfo8	Present domain-specific data in a clear, unobtrusive manner such that the information is tightly coupled to the environment and vice-versa	93	[Bowman et al., 1996]
SysInfo9	Strive for unique, powerful presentation of application-specific data, providing insight not possible through other presentation means	93	[Bowman et al., 1996]

4.3 VE User Interface Input Mechanisms

Table 10 Some Usability Issues of VE User Interface Input Mechanisms in General

VE User Interface Input Mechanisms in General

Label	*Usability Suggestion/Consideration*	*Page(s)*	*Bibliography Ref(s)*
Input1	Assess the extent to which degrees of freedom are integrabile and seperabile within the context of representative user tasks	96	[Jacob et al., 1994] [Zhai and Milgram, 1993b]
Input2	Eliminate extraneous degrees of freedom by implementing only those dimensions which users perceive as being related to given tasks	96	[Hinckley et al., 1994a]
Input3	Multiple (integral) degrees of freedom input is well-suited for coarse positioning tasks, but not for tasks which require precision	96	[Hinckley et al., 1994a]
Input4	When tasks require significant coordination and are not time critical (e.g., surgery), consider using "deviation in three-space" as a metric of device control (as opposed to time to target)	97	[Zhai and Senders, 1997]
Input5	From the user's perspective, device output should be consistent with, and cognitively connected to, user actions	97	[MacKenzie, 1995]
Input6	For fine positioning tasks, employ low gain, for gross positioning tasks, high gain. When VEs contain both coarse and gross positioning tasks strive for a balance between the two determined by iterative user testing of representative postioning tasks	98	[MacKenzie, 1995]
Input7	Address possible effects that prolonged usage with particular input device(s) may have on user fatigue and task performance	98	[Card et al., 1991] [Zhai, 1995]
Input8	Decrease user cognitive load by avoiding devices such as joysticks and wands which, in effect, place themselves between users and environments	99	[Davies, 1996]
Input9	Input devices should make use of user physical constraints and affordances	99	[Norman and Draper, 1986] [Hinckley et al., 1994a]
Input10	Avoid integrating traditional input devices such as keyboards and mice in combination with 3D, free-space input devices (devices that move freely with users, as opposed to mounted or fixed devices)	99	[Hinckley et al., 1994a]

Table 11 Some Usability Issues of Tracking User Location and Orientation

Tracking User Location and Orientation			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Tracking1	Consider the [Applewhite, 1991] framework when assessing the suitability of tracking technologies with respect to representative user tasks	101	[Applewhite, 1991]
Tracking2	Acoustic tracking technology is well-suited for multi-user systems where high data rates are needed and occlusion can be avoided	102	[Applewhite, 1991]
Tracking3	Magnetic tracking technology is well-suited for VEs with small working volumes and minimum electromagnetic interference	102	[Applewhite, 1991]
Tracking4	Mechanical tracking technology is well-suited for single user applications that require only a limited range of operation, applications where user immobility is not a problem	102	[Applewhite, 1991]
Tracking5	Optical tracking technology is well-suited for real-time applications where occlusion is less likely	102	[Applewhite, 1991]
Tracking6	When assessing appropriate tracking technology realitive to user tasks, one should consider working volume, desired range of motion, accuracy and prescision required, and liklihood of tracker occlusion	103	[Applewhite, 1991] [Waldrop et al., 1995] [Strickland et al., 1994] [Sowizral and Barnes, 1993]
Tracking7	Non-immersive, desktop VEs (e.g., VMDs) should employ head tracking techniques as a means to increase user presence by providing head motion parallax depth cues	104	[Barfield et al., 1997] [Hinckley et al., 1994a]
Tracking8	Head tracking can improve user presence and task accuracy of VMD visualization and manipulation tasks	104	[Barfield et al., 1997] [Snow, 1996] [Ware et al., 1993] [Rekimoto, 1995]
Tracking9	Head tracking is well-suited for search-based tasks	104	[Pausch et al., 1993]

TABLE 12 Some Usability Issues of Devices Supporting "Natural" Locomotion

Devices Supporting "Natural" Locomotion			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
NaturalLoc1	The walking in place metaphor is well-suited for VEs in which natural locomotion and a high sense of presence are required	105	[Slater et al., 1995a] [Templeman, 1997a]
NaturalLoc2	The walking in place metaphor may not be well-suited for VEs which require travel over large virtual distances (e.g., exploring extensive models)	106	[Slater et al., 1995a] [Templeman, 1997a]
NaturalLoc3	For natural locomotion where user roaming is not necessary, treadmill-based locomotion offers a simple, cost-effective solution	106	[Brooks et al., 1992] [Hirose and Yokoyama, 1992] [Virtual Space Devices, Inc., 1997]
NaturalLoc4	Supporting natural locomotion through devices such as treadmills and stationary bicycles may impose certain constraints on user movements	107	[Slater et al., 1995b]
NaturalLoc5	A constraint associated with treadmills is the fact that users must stand and walk in a very small, confined space	107	
NaturalLoc6	Be wary of any electro-mechanical device of significant size and power since an unexpected malfunction potentially places users at risk	108	
NaturalLoc7	Treadmills may not be well-suited for VEs which require travel over large virtual distances since users typically tire due to prolonged exercise or become frustrated with the slow pace imposed by most treadmills	108	[Brooks et al., 1992]

TABLE 13 Some Usability Issues of Data Gloves and Gesture Recognition

Data Gloves and Gesture Recognition			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Gloves1	Consider degrees of freedom issues discussed in Section 7.1 (they are directly applicable to glove-based input)	96, 109	[Jacob et al., 1994] [Zhai and Milgram, 1993b] [Hinckley et al., 1994a]
Gloves2	Understanding the complexity of representative glove-based tasks (with repspect to finger, thumb, and wrist flexion) may aid designers in identifying appropriate glove designs	109	
Gloves3	Very natural gestural interaction may be achieved through intuitive pinch mappings (e.g., *pinching with forefinger and thumb may used to grab a virtual object*, *snapping between middle finger and thumb*, etc.)	109	[MultiGen, 1997] [Fakespace, 1997]
Gloves4	Exoskeleton-based gloves are well-suited for very fine-grained manipulation tasks	110	[Exos, 1997]
Gloves5	Data gloves which have limited finger flex accuracy (5-10 degrees) may not be well-suited for complex recognition or fine manipulations	110	[Sturman and Zeltzer, 1994]
Gloves6	Exoskeleton gloves are not well-suited for general or casual use	110	[Sturman and Zeltzer, 1994]
Gesture1	To fully realize the power of natural gestural interaction, VE systems need to recognize a particular sequence of hand postures (tokens) as something more than simply the sum of the parts	111	[Sturman, 1992] [Su and Furuta, 1994]
Gesture2	Allow gestures to be defined by users incrementally, with the option to change or edit gestures on the fly	112	[Su and Furuta, 1994]
Gesture3	Image processing-based gesture recognition is well-suited for desktop or fishtank VEs	112	[Maggioni, 1993] [Bröckl-Fox et al., 1994]
Gesture4	Image processing-based recognition is more appropriate for single user VEs (since they typically require dedicated cameras and line of sight)	112	[Maggioni, 1993] [Bröckl-Fox et al., 1994]
Gesture5	Glove-based recognition systems are well-suited for VEs which allow some degree of roaming	112	[Wexelblat, 1995] [Jacoby et al., 1994]
Gesture6	Avoid gesture in abstract 3D spaces; instead use relative gesturing	112	[Hinckley et al., 1994a]

Table 14 Some Usability Issues of Magic Wands, Flying Mice, SpaceBalls, and Real-World Props

Magic Wands, Flying Mice, SpaceBalls, and Real-World Props			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
HandHeld1	Consider degrees of freedom issues discussed in Section 7.1 (they are directly applicable to hand-held input devices)	113	[Jacob et al., 1994] [Zhai and Milgram, 1993b] [Hinckley et al., 1994a]
HandHeld2	Free moving, isotonic input devices (such as tracked gloves) may be more useful when implemented as position controllers	115	[Zhai and Milgram, 1993b]
HandHeld3	Desktop, isometric input devices (such as the SpaceBall^{TM}) may be more useful when implemented as rate controllers	115	[Zhai and Milgram, 1993b]
HandHeld4	Elastic rate controllers are well-suited for object manipulation and positioning tasks	115	[Zhai and Milgram, 1994] [Zhai and Milgram, 1993a]
HandHeld5	Small, hand-held devices that exploit the bandwidth of human fingers may have performance advantages of devices relying on larger muscle groups, such as gloves	115	[Zhai et al., 1996]
HandHeld6	Assessment of six DOF input devices should include the degree to which size, shape, and use of device affords manipulation with fingers as opposed to larger muscle groups (e.g., wrist, forearm, shoulder)	115	[Zhai et al., 1996]
HandHeld7	Desktop, isometric devices are typically not worn, thus facilitating ease of device integration into working, desktop environments (e.g., CAD environments where users switch between six DOF device and keyboard, etc.)	116	
HandHeld8	Isometric input devices may be coupled with haptic feedback to facilitate bi-directional information flow between user and object, resulting in a more natural interaction	116	[Richard et al., 1996]
HandHeld9	Real-world props are an intuitive and powerful form of VE input (and output)	116	[Badler et al., 1986] [Hinckley et al., 1994b] [Stoakley et al., 1995]
HandHeld10	Real-world props allow the computer to interact with the real environment controlled by the operator	116	[Badler et al., 1986] [Hinckley et al., 1994b]
HandHeld11	Real-world props or tools with mass enable bi-directional information flow inherent in complex user-VE interaction	116	
HandHeld12	Fatigue associated with use of real-world props may be reduced by including some type of clutching mechanism or by providing proper arm support	116	[Stoakley et al., 1995]

TABLE 15 Some Usability Issues of Speech Recognition and Natural Language Input

Speech Recognition and Natural Language Input			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Speech1	Combining speech input and pointing may result in more usable selection mechanism	120	[MacKenzie, 1995] [Hauptmann, 1989]
Speech2	Use speech recognition and natural language input as a complement to multimodal interfaces, as opposed to stand-alone mechanisms	120	[to the Interface, 1996] [Hauptmann, 1989] [MacKenzie, 1995]
Speech3	Natural speech recognition can make VEs easier to use be offering the ability to make more direct charges to the environment	121	[Karlgren et al., 1995]
Speech4	Include a proxy, agent, or god-like entity for users to verbally address	121	[Karlgren et al., 1995] [SRI, 1996]
Speech5	Strive to support *incremental* human-computer discourse	122	[Cohen, 1992] [Karlgren et al., 1995]
Speech6	Sophisticated natural language recognition systems should "learn" user syntax and semantics so that the computer interprets user language (as opposed to the other way around)	122	[Karlgren et al., 1995]
Speech7	Verbal annotation is useful for applications areas, such as visualization, simulation, and training VEs, where preserving contextual information is important	123	[Harmon et al., 1996]
Speech8	Strive for seamless integration of annotation into VEs, requiring no mode switching to record	123	[Verlinden et al., 1993] [Harmon et al., 1996]
Speech9	When designing voice-based annotation, provide quick, efficient, and unobtrusive means to record and playback annotations	123	[Verlinden et al., 1993] [Harmon et al., 1996]
Speech10	When designing voice-based annotation, allow users to edit, remove, and extract or save annotations	123	[Verlinden et al., 1993] [Harmon et al., 1996]

4.4 VE User Interface Presentation Components

Table 16 Some Usability Issues of Visual Feedback — Graphical Presentation

Visual Feedback — Graphical Presentation			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Visual1	Use stereopsis when information is presented in an egocentric view	126	[Davis and Hodges, 1995] [Drascic, 1991]
Visual2	Use stereopsis when monocular cues are ambiguos or less effective than stereoscopic cues	126	[Davis and Hodges, 1995]
Visual3	Use stereopsis when presenting relatively static scenes	126	[Davis and Hodges, 1995] [Drascic, 1991]
Visual4	Use stereopsis when presenting complex scenes, unfamiliar, or ambiguous objects	126	[Davis and Hodges, 1995] [Drascic, 1991]
Visual5	Use stereopsis when 3D manipulation tasks require ballistic movements	126	[Davis and Hodges, 1995] [Drascic, 1991]
Visual6	Use stereopsis when user tasks are highly spatial (e.g., precise placement of tools, 3D docking, visual searching)	127	[Davis and Hodges, 1995] [Drascic, 1991] [Barfield et al., 1997]
Visual7	Couple user field of view and display via head tracking	127	[Pausch et al., 1993]
Visual8	Strive for high refresh and update rates to minimize latency or lag	128	[Richard et al., 1996] [Ware and Balakrishnan, 1994]
Visual9	For 3D target acquisition tasks, use input devices which have low lag, ideally less than 50msec	128	[Ware and Balakrishnan, 1994]
Visual10	For 3D target acquisition tasks, separate head lag from land lag	128	[Ware and Balakrishnan, 1994]
Visual11	For 3D target acquisition tasks, decouple the target and cursor from the rest of the environment, so that higher update rates can be applied to the target and cursor only	129	[Ware and Balakrishnan, 1994]
Visual12	Consider how representative user tasks implicitly suggest mix of immersion, self presence, and object presence required (and subsequently display type)	130	[Bennett et al., 1996]
Visual13	HMDs are best-suited for single, autonomous user activity	130	[Bennett et al., 1996]

Table 16 Some Usability Issues of Visual Feedback — Graphical Presentation (cont'd)

Visual Feedback — Graphical Presentation			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Visual14	Eliminate obvious interface boundaries, strive for seamless user inclusion	131	[Bricken, 1990]
Visual15	HMDs are well-suited for applications where complete visual immersion or absence of distractions is required	131	
Visual16	HMDs are typically tethered by audio and video cabling, limiting user mobility to cable length and support mechanisms	132	[Kocain and Task, 1995]
Visual17	BOOM^{TM} and PUSH^{TM} Desktop Displays can be seamlessly integrated into user work activity, exploiting existing user work habits	133	[Fakespace, 1997]
Visual18	Spatially Immersive Displays are well-suited for spatially rich applications	135	[Bennett et al., 1996]
Visual19	Use Spatially Immersive Displays when an enormous field of view is required	136	[Bennett et al., 1996]
Visual20	For multi-user tasks and collaboration consider Spatially Immersive Displays and Virtual Model Displays	136	
Visual21	Spatially Immersive Displays are not well-suited for multi-user VEs that require separate images per user	137	
Visual22	Virtual Model Displays are particularly well suited for providing exocentric views of virtual models	138	[Bennett et al., 1996] [Naval Research Laboratory, 1997]
Visual23	Virtual Model Displays are well-suited for local collaboration, since multiple users can participate using the single display	139	[Bennett et al., 1996] [Naval Research Laboratory, 1997]
Visual24	Virtual model shape and size as well as desired user point of view may suggest Virtual Model Display pitch, and subsequently type	140	[Bennett et al., 1996]
Visual25	Virtual Model Displays are particularly well-suited for model prototyping and other tasks which require manipulation of some external model	141	[Bennett et al., 1996]

TABLE 17 Some Usability Issues of Aural Feedback — Acoustic Presentation

Aural Feedback — Acoustic Presentation			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Aural1	Use aural feedback effectively to improve user performance of tasks, such as three-dimensional target acquisition and shape perception	142	[Mereu and Kazman, 1996] [Mereu, 1995] [Hollander and Furness, 1994] [DiGiano and Baecker, 1992]
Aural2	Present aural information in a meaningful, timely, and useful manner	142	[Cohen and Wenzel, 1995]
Aural3	When appropriate, associate audio "direction" with content and other attributes such as pitch, duration, and meter	143	
Aural4	Support visually impaired users through rich audio and haptic feedback	38, 80, 144	[Mereu and Kazman, 1996]
Aural5	Provide three-dimensional audio feedback when detailed separation, isolation, position, or spatial/directional content are required	145	[Cohen and Wenzel, 1995]
Aural6	Strive to generate real-time sounds (to accentuate users' *actions*, *observations*, and *experiences*)	145	[Cohen and Wenzel, 1995]
Aural7	Provide high bandwidth aural channels to support simultaneous, dynamic presentation of many different sounds, from many different locations, at varying intensity levels	146	[Cohen and Wenzel, 1995]
Aural8	For same-site, multi-user VEs, choose loudspeaker audio over headsets	146	
Aural9	Use headsets for a portable, cost-effective audio system for remote single users	147	

Table 18 Some Usability Issues of Haptic Feedback — Force and Tactile Presentation

Haptic Feedback — Force and Tactile Presentation			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Haptic1	Haptic presentation is effective in areas where other senses may not be usable	147	[Kaczmarek and Bach-Y-Rita, 1995]
Haptic2	Use other sensory information to reinforce or enhance haptic tasks	148	[Burdea and Coiffet, 1994]
Haptic3	Effective VE haptic displays should include devices which provide both kinesthetic and tactile information	148	[Kaczmarek and Bach-Y-Rita, 1995]
Haptic4	Use vibratory cues as they inherently exist in real-world tasks (not simply as *generic* tactile cues)	148	[Kontarinis and Howe, 1995]
Haptic5	When possible, present kinesthetic and tactile cues separately	149	[Kaczmarek and Bach-Y-Rita, 1995]
Haptic6	Ensure high bandwidth force reflection with high stiffness between master and slave devices	149	[McNeely et al., 1995]
Haptic7	Use haptic devices to provide strength and speed for natural end use	149	[McNeely et al., 1995]
Haptic8	Use haptic devices to present high resolution force and position to users	150	[McNeely et al., 1995]
Haptic9	Use haptic devices to provide reliable, intuitive, low fatigue operation	150	[McNeely et al., 1995]
Haptic10	Support high bandwidth haptic interaction	150	[Brooks et al., 1990] [Hannaford and Venema, 1995]
Haptic11	Be wary of complex, multi-degrees-of-freedom haptic systems	150	[McNeely et al., 1995]
Haptic12	Tactile displays need not provide incredibly high (spatial) resolution	151	[Kaczmarek and Bach-Y-Rita, 1995]
Haptic13	Avoid simultaneous haptic presentation of complex patterns, sensations, or objects (instead, consider tracing or edge enhancement)	151	[Kaczmarek and Bach-Y-Rita, 1995]
Haptic14	Be cautious in presenting, and semantically binding a large number of haptic intensity levels	151	[Kaczmarek and Bach-Y-Rita, 1995]

Table 18 Some Usability Issues of Haptic Feedback — Force and Tactile Presentation (cont'd)

Haptic Feedback — Force and Tactile Presentation			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
Haptic15	Maximize user comfort and ease of use (e.g., *minimizing* user grasping-force, *optimizing* work-to-rest ratio)	152	[Wiker et al., 1991]
Haptic16	Hand-worn devices should be lightweight, unencumbering, and portable enough to allow users sufficient freedom of motion	152	[Hannaford and Venema, 1995]
Haptic17	Hand-worn devices (as well as other haptic devices) should allow uninhibited, effortless motion when *no* virtual forces exist	152	[Hannaford and Venema, 1995]
Haptic18	For mechanical exoskeletons, ensure that joints of the device fit joints of the user, so that body movement is natural and not forced	153	[Zyda et al., 1995]
Haptic19	Base haptic device design or purchasing decisions on the nature and frequency of representative user tasks	153	[Shimoga, 1993]
Haptic20	When possible, provide kinesthetic or tactile feedback for manipulation-based tasks	154	[Kontarinis and Howe, 1995]
Haptic21	Non-body-centered devices (such as joysticks) may hamper interaction between VE and user by the fact that users manipulate *external* devices	155	[Davies, 1996] [Jacobsen, 1996] [Bricken, 1990]
Haptic22	Joysticks and specialized devices are better suited for relatively stationary tasks (where users do not move about)	155	
Haptic23	When appropriate, integrate "tools with mass," to provide users with the same natural, gravitational, and inertial kinesthetic feedback in the VE as they experience in the real world	156	[Hinckley et al., 1994a] [Stoakley et al., 1995]

TABLE 19 Some Usability Issues of Environmental Feedback and Other Presentations

Environmental Feedback and Other Presentations			
Label	***Usability Suggestion/Consideration***	***Page(s)***	***Bibliography Ref(s)***
MiscCues1	Provide additional sensory information other than the "big three": visual, aural, and haptic	157	[Carter, 1992] [Barfield and Danis, 1996]
MiscCues2	When possible, use olfactory information to provide additional directional and distance cues (e.g., in conjunction with visual and/or aural cues)	158	[Barfield and Danis, 1996]
MiscCues3	Integrate and present environmental cues and their effects (e.g., temperature, wind, rain) in full force to increase the believability, usability, and training transfer associated with the VE	158	

5 Users and User Tasks in VEs

At the root of all insightful usability evaluations are application-specific, representative user tasks. It is meticulous examination of user performance and satisfaction, physical device support, and software facilities in support of users' cognitive organization of these tasks which not only expose critical usability problems, but promise the most notable improvements when addressed.

The growth in interest of user-centered design has placed well-deserved importance of user requirements and task analysis on the interaction development cycle [Hix and Hartson, 1993]. Indeed, "the entire interaction development cycle is becoming centered around the evaluation of users performing tasks"[Hix and Hartson, 1993]. As a result, the *user task* has become a basic element in usability engineering.

Given the widespread applicability of VEs, their potential user task space is enormous. However, a thoughtful approach to understanding a smaller, yet representative, subset of this space may be helpful. Identifying basic task characteristics and elementary VE tasks representing some "common denominator" may be appropriate, since any findings at this level could presumably be applied in a more application-specific or high-level task analysis.

The following three sections provide a starting place for enumerating and understanding usability issues of VE-based tasks. We first take a look at some characteristics of user tasks in general, followed by a more detailed look at the types of low-level tasks typically found in VEs. Last, we examine a possible relationship between the low-level tasks discussed, task sets, and application areas of VEs.

5.1 Characteristics of Users and User Tasks in VEs

In a 1996 *Virtual Reality Annual International Symposium* article, Kay Stanney states that "one important aspect that will directly influence how effectively humans can function in virtual worlds is the nature of the tasks being performed" [Stanney, 1995]. In determining the nature of user tasks, Stanney suggests that some tasks may be uniquely suited to virtual representation while others may simply be impractical. Understanding the relationship between real-world task characteristics and their corresponding virtual task characteristics is key in determining how well a task is suited for VEs.

A key question is, then, *which* task characteristics determine whether a particular task is appropriate for a VE? Some of the most frequently cited objective measures of task performance are task completion time, task error rate, and task learning time [Hix and Hartson, 1993]. Thus it seems reasonable to address characteristics which have significant effects on these measures. One approach is to look at task characteristics which describe *who* is performing the task and *where* the task is being performed, as well as characteristics inherent in the basic components of tasks.

Which usability characteristics have the most significant effect on task performance?

5.1.1 User Differences and Demographics

It is widely accepted that differences among users have a profound effect on task performance. Identifying these differences and their implications is important if VEs are to effectively accommodate many types of users. Fortunately for VE researchers, user differences have been well studied in the area of HCI.

For instance, ***user experience*** ≪Users1≫ has been shown to have a direct impact on user skills and abilities normally associated with task performance. User experience also affects the manner in which users understand and organize task information [Egan, 1988]. A user new to VEs may be able to apply traditional computer experiences within the VE to improve task performance (e.g., working with menus). However, direct VE experience gives a user familiarity with VE-specific issues such as field of view, suspension of belief, stereoscopic vision, and even motion sickness.

Domain knowledge ≪Users2≫ is another type of user experience to consider. Identifying the type and complexity of a typical user's domain knowledge helps in developing the type and complexity of information in a VE. In short, VEs should be powerful enough to allow for productive, expert work while being simple enough to allow for novice exploration and learning.

Other user characteristics related to ***technical aptitudes*** also appear to have a significant effect on performance [Egan, 1988]. Such aptitudes include ***orientation***, ***spatial visualization***, and ***spatial memory*** ≪Users3≫. According to Egan, users with low spatial abilities generally have longer task performance times with more errors on first trial. Many of these users find themselves lost within the system, suggesting that their low spatial abilities have a direct, and in this case negative, influence on their navigating abilities. In comparison to 2D interface users, VE users rely more heavily on their own spatial propensities. Stanney states that VEs must address the issue of assisting spatially challenged users with spatial orientation. Solutions include developing better design metaphors and providing additional navigational information such as context, landmarks, and maps [Stanney, 1995] [Darken and Sibert, 1995]. For example, the World in Miniature (WIM) metaphor developed at the University of Virginia allows the user to gain a larger context of the environment by providing a miniature version of the VE as a dollhouse within the VE [Stoakley et al., 1995]. Spatially useful navigational information may include virtual landmarks or simply a map.

VE systems should ***support both right and left-handed users*** ≪Users4≫ by providing symmetric or interchangeable input devices. Placement of command menus within the VE should be taken into consideration. For example, when using a "painter's palette" metaphor for command menu interaction, the virtual palette should "appear" in the correct hand, depending on user preference. Accommodating natural interaction for both right- and left-handed users is discussed in more detail in Section 5.2.3.

Other user characteristics such as ***age*** and ***gender*** ≪Users5≫ may affect task performance by predicting influential characteristics such as experience and technical adeptness. In an increasingly accommodating world, VEs should be able to adapt to both physically and mentally challenged users, through for example, device choice, available modes of feedback, and physical layout of workspace. Other physical characteristics such as ***size*** and ***stature*** ≪Users5≫ may only come into play when a user must wear VE equipment. For example, it is more difficult for a child to perform a complex grasping task with an adult-sized data glove than a child-sized data glove. Similar problems exist in head-mounted display (HMD) design. Kaiser Electro-Optics, a leading HMD manufacturer, claims that their helmets are designed to fit 95% of users — a requirement by customers, such as the military, who have many different users training with or utilizing a single HMD [Kaiser Electro-Optics, 1996]. User stature and size also effect the user's extent of reach, length of step, and strength of grasp. For example, researchers at the University of Washington found their design of "virtual Seattle" to be ill-suited for children, as the 3D menus were displayed at an adult's shoulder height. To select a menu item, children were having to "fly" up to the menu, and then attempt to "stop flying" at the adult-tailored menu height [University of Washington, 1996]. Cognitively, the children knew exactly what to do. Physically, the task was unnecessarily frustrating. One solution is to allow users to "wear" different sized virtual bodies. Boeing used such an approach in the design of the Boeing 777, thus allowing designers to get an idea of how well the airplane would accommodate persons of varying stature [Boeing, 1996].

5.1.2 Number of Users, Location of Users, and Collaboration

The ***number*** and ***location of users*** ≪Tasks1≫, coupled with the nature and intent of user tasks, must be taken into consideration when assessing the usability of VEs. Many VE interfaces are designed for and restricted to single, autonomous users. More recently, the value of collaborative and sometimes remote work has started to receive attention in VE research. To support these types of interactions, researchers not only need to re-evaluate typical tasks and use of input and output devices, but also to ***integrate socially-minded considerations such as group communication, role-play, and informal interaction*** ≪Tasks2≫ — considerations well studied and addressed in current computer-supported cooperative work (CSCW) journals. Such considerations were made during Mitsubishi's Electronic Research Lab's development of "Diamond Park", a socially constructed VE containing elements of real-world parks where people from geographically distinct locations can come together to interact [Waters et al., 1997].

Usability characteristics associated with single-user VEs are similar to those of single-user GUIs. That is, users are typically focused on a single task, interacting with a simple set of hardware devices. Matches between hardware and tasks are somewhat easier to infer, since interaction sequences in single-user VEs are more tractable and more common than multi-user systems. Users are able to cognitively attribute system reactions to a consequence of either their own or system action. There is essentially no social interaction required. Some existing VE hardware is biased toward single user scenarios. For example, by design, HMDs, audio headsets, and force-feedback joysticks accommodate one user at a time. Attempting to support more than one user entails not only supplying each user with a full complement of interface mechanisms, but managing each user's interface with respect to all other users.

Multi-user VEs imply a more complex arrangement of user tasks, interaction, and hardware. As such, an understanding of interrelationships between users and tasks is essential. In a cooperative setting, tasks are typically multi-threaded and highly interrelated as users work together toward some common goal. In light of this, cooperative multi-user VEs should support ***social organization***, ***construction***, and ***execution of plans*** ≪Tasks3≫. For example, VEs may support social organization by facilitating role appointment via audio discussion channels and visual labeling of roles (e.g., textually or iconically). VEs may support social construction by allowing users to discuss, distribute, and allocate tasks, perhaps in the form of a virtual "to do" list or notepad. And finally, support for performance of tasks involves allowing users to perform autonomous, concurrent tasks.

A good example of VE hardware well-suited for collaborative work is the Responsive Workbench [Naval Research Laboratory, 1997]. Much like a large table, users are able to stand around the Workbench interacting with common virtual artifacts using a hand-held "wand". Since users are not immersed in HMDs, verbal and gesture interaction are practically no different than that experienced in real-world interaction. Social roles are well supported as "leaders" may locate themselves in central positions along the Workbench edge. With the use of complementary input devices, such as multiple pointers, and auxiliary displays, such as virtual binoculars or a BOOM™ (see Section 8.2.1, users can simultaneously explore autonomous yet related tasks. For example, as "leaders" survey an exocentric, large-scale view of a battlefield presented on the Workbench surface, another user may use virtual binoculars to scout an egocentric view atop a strategic mountaintop.

In competitive multi-user systems, there may be less importance placed on organization of group decisions and execution, and more importance placed on providing ***awareness-based information*** ≪Tasks4≫ such as that critical to each user's current task and that which suggests a relative ranking among users. For example, in a tank battle simulation, competing users require task-relevant information such as their current position, orientation, and weapons supply. Yet they also require competitive-based information such as "how many tanks do other users have."

The location of users in a multi-user VE may also affect usability concerns. When users are collocated, certain types of interaction become unmanageable or even counter-intuitive. For example, a room full of users individually equipped with HMDs can be computationally expensive — resulting in system lag. Compound this with "hobbled" social interaction implicit in communication through virtual avatars and it becomes likely that HMDs are less well-suited for collocated, collaborative work. Compare this to the Responsive Workbench example given above, where there are no virtual avatars; each participant is his or her own real-life "avatar." On the other hand, if collaborating users are each located in different locations, it is unlikely that each will have access to a networked Workbench or Cave Automatic Virtual Environment (CAVE™) (discussed in Section 8.2.1). In this case, HMDs may be a better solution, allowing all users to work in a common, virtual space. Some other concerns related to distributed, cooperative VEs, along with overviews of existing schemes for distributed VEs, is given in [Broll, 1995].

In general, if users physically occupy a single space, it may not be necessary to synthesize individual views and experiences. When users do not physically occupy a single space, it is usually desirable to synthesize and facilitate access to, and understanding of, this space.

5.1.3 Temporal Aspects of Tasks

After identifying key tasks undertaken by users of VEs, a reasonable next step is to understand how these tasks are temporally related. We consider three temporal relationships among tasks:

- *concurrent* and *interleavable* — tasks performed at the same time,
- *serial* — tasks performed in distinct sequence, and
- *stackable* — tasks which, once started, lead to other tasks integral in the completion of earlier tasks.

This type of analysis can be performed on many levels, depending upon factors such as the number of users and types of tasks. For example, in a single-user scenario it is useful to understand which tasks a user may want or need to perform ***concurrently*** ≪Tasks5≫. At a lower level, each of these tasks may involve a set of ***serial*** ≪Tasks6≫ subtasks. In a multi-user scenario, support and analysis of concurrent, serial, or stackable tasks may be performed at a higher level, focusing more on the relationships among all users' tasks than on an individual's tasks.

In the real world, humans naturally perform several concurrent tasks in support of a common goal. For example, when driving a car, "users" concurrently operate an accelerator or brake, steering wheel, and air-conditioner. Cognitively, users are also orienting, assessing current air temperature for comfort, and deciding what to eat for dinner. Supporting user multitasking is generally accepted as useful. Facilitating user instantiation, monitoring, completion, and integration of concurrent tasks in VEs should allow more experienced users to perform complex tasks in less time.

Identifying serial tasks in a VE may help designers assess and subsequently recommend more useful interaction techniques. A good example can be found in interior or architectural design applications. Consider the basic task of designing a three dimensional office space. As part of this task, a possible serial sequence of subtasks may be first populating the space with furniture, and then arranging and rearranging the furniture within the space. In this case, the iterative selection of furniture items should be intuitive, requiring minimal effort and repetition of command sequences. For example, a "furniture palette" which remains open until explicitly closed, and which places selected items at the user's virtual feet, allows for ease of iterative selection and subsequent placement of furniture. On the other hand, an interaction requiring users to first open the palette, select a piece of furniture, release the furniture from grasp (selection of object infers possession), and then re-open the automatically closed palette does not support this type of serial task ordering. Quite the contrary, the latter interaction supports just the opposite task ordering; iteratively select then place each piece of furniture. It is unreasonable to assume that developers will know which serial sequence is "the correct one." It is only through user-centered processes such as participatory design and usability evaluation that such distinction can be effectively made.

Some user scenarios may involve a sequence of stackable subtasks, in which case interfaces should support a type of "navigation" among the subtasks. The "back" and "forward" buttons in most World Wide Web (WWW) browsers provide such functionality to the task of perusing hypertext documents. In a VE, support for ***stackable*** $\ll$Tasks7$\gg$ tasks may aid users in error recovery. For example, in a training simulator, critical errors and the sequence of decisions and subtasks leading to the errors may be "stepped though" to review where faulty decisions and actions were made. Another example is the simple inclusion of an "undo" command. Support for navigation through a stackable sequence of tasks involves storing a history of actions, which in turn may be expensive. Thus, identifying and prioritizing stackable tasks may be desirable.

5.2 Types of Tasks in VEs

As previously mentioned, one approach to understanding the enormous task space inherent in VEs is to examine general, representative task spaces which, in effect, serve as a "common denominator" for further analysis. [Esposito, 1996] identifies five such task spaces, or "interaction areas" as:

- *navigation*,
- *object query* (for purpose of selection and information),
- *object manipulation*,
- *object creation* and *modification*,
- *application environment query* and *modification*.

To simplify matters, we consider object creation and modification to be specialized forms of object manipulation. That is, to create an object, a user typically has to manipulate some other object (e.g., menu, palette, toolbox). We also consider the VE application to be a specialized instance of an object. Thus, application environment query and modification are discussed in terms of object query and modification. We acknowledge the fact that the distinctions made by [Esposito, 1996] warrant further investigation, and in turn will allow for more specialized, distinct discussion of VEs.

Thus, for purposes of our taxonomy, we consider the following three general task spaces:

- *navigation* and *locomotion*,
- *object selection*, and,
- *object manipulation*, *modification*, and *query*.

5.2.1 Navigation and Locomotion

It is human nature to explore the world around us, from an infant's exploration of a small crib to an adult's exploration of a vast mountainside. From a very early age we instinctively learn to walk, and subsequently navigate, through the world around us. The amount of cognitive and psycho-motor activity required for human navigation and locomotion is indeed significant, although for humans, it is second nature. As such, implementing navigation and locomotion mechanisms for, and supporting interactions within, VEs is a significant usability issue. As such, it remains an open research issue in current VE literature.

One approach to understanding navigation and locomotion requirements in VEs is to consider cognitive issues related to navigation. For example, [Darken and Sibert, 1996] presents a classification for wayfinding (navigation-based) tasks that identifies key design issues related to user cognition during wayfinding. Their classification breaks wayfinding tasks into three primary categories ≪Nav1≫:

1. ***Naive Search***: Searching tasks where users have no prior knowledge of target location;
2. ***Primed Search***: Searching tasks where users know location of target *a priori*; and
3. ***Exploration***: Wayfinding tasks where there are no targets.

As [Darken and Sibert, 1996] points out, purely naive searches are rare in the real world, as most people have some idea of at least localized space around them. In virtual spaces however, users are typically introduced into worlds never seen; thus performance in VEs hinges on users' ability to quickly comprehend spatial relationships. To support user tasks in newly experienced or complex environments, VEs should support user exploration, understanding, synthesis, and annotation of space.

For example, the ***layout***, or ***floor plan, of a VE should be consistent*** ≪Nav2≫. A consistent setting may increase a user's ability to conceptualize the space as a whole as users make use of object location and inter-object distances to encode topological or *survey knowledge* [Darken and Sibert, 1996]. This knowledge, in turn, has been shown to be essential during navigation and wayfinding [Lynch, 1960]. Darken and Sibert [1995] present a set of ***organizational principles*** ≪Nav3≫ which support a user's mental organization of VEs when wayfinding. These principles include:

1. ***Dividing a large world into distinct smaller parts***, where the division should be hierarchical in nature;
2. ***Structuring the smaller parts under a simple organizational scheme*** such as a grid, or logical spatial ordering such as a street naming convention; and
3. ***Providing frequent directional cues***, such as landmarks or an on-screen compass.

Principles 1 and 2 suggest that the VE itself contains information specifically for spatial organization, such as a grid overlaid upon the environment, or ***spatial labels*** ≪Nav4≫ (e.g., street names) at appropriate locations. ***Landmarks*** ≪Nav4≫, such as notable or prominent structures, have also been shown to be useful in wayfinding tasks [Darken and Sibert, 1995]. The consistent placement, appearance, and functionality of such landmarks may be important in providing a coherent spatial frame of reference. When appropriate, a visible ***horizon*** ≪Nav4≫ is also an important environment component, providing the user with a spatial grounding and a sense of elevation [Bennett et al., 1996].

Another approach to understanding navigation and locomotion requirements in VEs is to consider essential questions posed by those navigating in VEs. For example, [Wickens and Baker, 1995] asserts that "appropriate implementation of the human factors of navigation assist the user to answer four (key) questions ≪Nav5≫:

- ***Where am I now?***
- ***What is my current attitude and orientation?***
- ***Where do I want to go?***
- ***How do I travel there?***

The first three questions are supported by inclusion in the VE of navigational aids such as maps, landmarks, horizons, and grids as mentioned above. Answering the question "How do I travel there?" is particularly challenging for VE developers and users, since most physical constraints that normally accompany real-world navigation and locomotion are not present [Wickens and Baker, 1995], while other physical constraints not found in the real world (e.g., user-worn VE equipment) may be present.

Navigation and Locomotion Metaphors

To minimize a user's cognitive load during VE task performance, navigation metaphors are typically used. Use of appropriate metaphors in user interface design is generally considered to be good practice, since they can effectively exploit users' prior knowledge to increase familiarity of action, procedures, and concepts [Neale and Carroll, 1997]. Another advantage of metaphors is that they help maintain consistency across interface tasks; walking in one part of the VE should be performed no differently than walking in another part of the VE. On the other hand, a poor navigation or locomotion metaphor may create a number of problems for the user. For example, metaphors which have a number of *mismatches* (semantic differences between real and virtual worlds) may leave users confused about available functionality and mappings. Similarly, restricting design to a single metaphor places inherent limitations on potential related user interactions; for example, under a "walking" metaphor, users should not be able to "fly."

[Ware and Osborne, 1990] classifies control metaphors for 3D interaction into the following areas:

- *Eyeball-in-hand* — user viewpoint is controlled via direct manipulation of virtual camera,
- *Scene-in-hand* — exocentric user view is used to directly manipulate virtual objects, and
- *Flying vehicle control* — user "flies" a vehicle to navigate through a VE.

The scene-in-hand metaphor addresses object manipulation more so than navigation. However, if the virtual object represents the user, then the metaphor can be used quite effectively for navigation (as in the case of WIM, discussed below). Note that control metaphors such as "walking in place" [Slater et al., 1995a] [Templeman, 1997a], [Virtual Space Devices, Inc., 1997] are not accounted for in this classification. Thus we can add a fourth item to the classification:

- *Real world control* — walking, walking-in-place, treadmills

Locomotion via treadmills and walking in place is discussed further in Section 7.2.2.

The most common VE metaphor for locomotion is based on flying vehicle control. These VEs typically rely on hand-based gestures and orientation of hand-held pointing devices to determine direction and velocity [Slater et al., 1995b]. One reason hand-based metaphors for locomotion are so popular may be because no *physical* locomotion is required; users can travel arbitrary distances without leaving their seat, or walking at all.

[Fairchild et al., 1993] created a "flying hand" metaphor in which users pressed a button mounted on a tracked spatial device to initiate flying. Once pressed, the position of the user's hand relative to their head determined the direction and velocity of travel. Informal user observations revealed that "HMD-clad users have a difficult time finding buttons on hand positioning devices" [Fairchild et al., 1993]. Thus, the metaphor was modified to a "floating guide," the only difference being that users pressed a virtual "button" (which remained floating next to a user's head) instead.

Under what conditions would mode-based interaction be desirable?

When a locomotion metaphor, such as the "hand-gesture" metaphor mentioned above, is presented though a hand-held device or hand-based tracker, inherent restrictions are placed on user activity. In particular, using one's hand for locomotion purposes makes that hand unavailable for any dexterous, hand-based manipulation tasks. In short, users are forced to perform *either* a navigation task *or* a manipulation task, mutually exclusive of each other. Restricting user activity during locomotion tasks may have negative effects on user performance and acceptance. Instead, VEs should strive to support concurrent and consistent tasking (see Section 5.1.3). [Fairchild et al., 1993] suggests that VE developers ***avoid mode-based navigation*** ≪Nav6≫, where users cannot perform any other tasks but locomotion. Observations show that users approaching objects of interest often overshoot or undershoot, requiring a mode shift back to "navigation mode" to perform location adjustments. Once adjusted, users would then have to switch modes once again to manipulate or query the object of interest. It comes as no surprise that users found this type of moded interaction frustrating and counter-productive.

On the other hand, one type of ***body-centered interaction*** ≪Nav7≫ supports locomotion through our natural means of locomotion: walking [Slater et al., 1995b]. Tracked users walk about a physical space that represents a virtual space. Magnetic tracker range restrictions (see Section 7.2.1), facility space limitations, and application domain may force the physical space available to be much smaller than the space it needs to represent (i.e., the virtual space). In these cases, users must have some means (other than a simple 1:1

mapping of physical to virtual space) of locomotion available. Many VEs use *walking in place* to solve space limitations, allowing users to travel arbitrary distances and directions. Typically the metaphor uses torso orientation to determine direction and walking rate to determine velocity; thus it is much like walking in the real world. An advantage of this type of body-centered interaction is that it does not force mode-based performance of tasks; that is, user hands are free to perform any manipulation tasks normally associated with hands while walking.

A similar body-centered interaction technique uses *leaning* as a locomotion metaphor. [Fairchild et al., 1993] describe the leaning metaphor as "stunningly effective as a navigation paradigm for reasonable complex spaces." Modeled after humans' natural tendency to lean towards persons or objects of interest, this technique uses *direction of lean* for virtual direction and *distance of lean* for velocity. [Fairchild et al., 1993] studies showed that users typically overshoot the target while learning to use the leaning metaphor, but soon learn to use it effectively.

[Fairchild et al., 1993] also makes an interesting distinction between *absolute leaning* and *relative leaning*. Absolute leaning uses head displacement to determine distance, so that users lean forward to move forward, and backward to move backwards. While absolute leaning in this form may be suited for VEs in which users remain spatially centered, it seems that the technique would be particularly tiresome for users who wish to explore objects and terrain outside their current location. Relative leaning uses head displacement to determine velocity, so that users are able to move from point to point with no notion of a "centralized" location. Although [Fairchild et al., 1993] claimed that relative leaning was useful for movements across long distances, it was not clear how long users are physically able to lean with any amount of accuracy.

The lean metaphor is similar that used in Osmose, an abstract VE developed at Soft Image [Davies, 1996]. In Osmose, users navigate under a "scuba diving" metaphor, breathing in to rise, and out to fall. Figure 2 shows an immersed developer of Osmose. Note the hardware worn across the chest (for measuring expansion of the chest).

Trackers on the HMD and waist are used to determine lean direction. Although studies have shown this metaphor to be well received by most users, the fact that not everyone is familiar with scuba navigational methods implies that some users must learn how to navigate the VE from scratch.

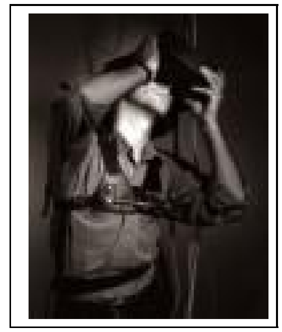

FIGURE 2: Osmose's Unique Navigation Gear [http://www.softimage.com/]

As [Wickens and Baker, 1995] points out, the effectiveness of a navigation metaphor may be a function of user task and application environment. Indeed, most approaches to navigation and locomotion work well in some cases, but not in all [Fairchild et al., 1993]. It is important that ***interface metaphor(s) intuitively match user-task space*** ≪Nav8≫ as well as allow for ***concurrent task execution*** ≪Nav9≫. As pointed out earlier, a poor interface metaphor may cause more problems than it solves, introducing new ambiguities to users who struggle to understand subtle differences between real-world and virtual knowledge.

Point-to-Point Navigation

Some VEs allow users to identify target locations through abstract means. This type of navigation is termed *point-to-point* navigation. Many of the usability issues associated with point-to-point navigation revolve around two central issues: the manner in which users specify target locations and the manner in which users are virtually relocated.

Specifying a new location can be implemented in a number of ways including:

- *descriptive specification* — e.g., move to "home",
- *relative specification* — e.g., move 100 feet forward,
- *technical specification* — (x,y,z) coordinates, latitude/longitude pair, and
- *specification via manipulation of representative virtual objects* — relocate one's avatar from an exocentric point of view.

An interesting implementation of the scene-in-hand metaphor, discussed previously, is the WIM developed at the University of Virginia [UVA, 1996]. The WIM is a miniature representation of the VE. When users move objects in the WIM, the corresponding virtual object is moved [Pausch et al., 1996]. If desired, the WIM can contain a graphical representation of user position and orientation (typically a camera and viewing frustum) that can be manipulated to specify new user locations and orientations.

When specifying new locations via virtual object representation, the immersive world must be updated, at some point, to reflect the specified change. Many point-to-point navigation schemes use animated user viewpoints to move users from an initial to a specified location. For example, initial WIM designs automated navigation by "flying" users through the environment. However, studies showed that users "cognitively vested" themselves in the miniature, keeping focus on changes in the WIM as opposed to the immersive environment [Pausch et al., 1996]. The design was modified to accommodate user preferences by animating locomotion via flight into the WIM. That is, the animation sequence flew users into the WIM's camera icon so that the WIM becomes the new immersive environment. When the specified location is reached, the old full-scale world is faded, revealing the new WIM.

In some cases, automating point-to-point navigation may cause problems, since users glean limited spatial knowledge from the experience. Users may be shown, and subsequently may learn, little or nothing about virtual terrain between initial and target locations. Thus, point-to-point navigation in VEs is essentially a tradeoff between ***speed and flexibility of navigation*** versus ***situational awareness*** ≪Nav10≫ [Wickens and Baker, 1995]. In cases where users specify target location via some virtual object, particularly as in the case of a WIM, users are given some contextual information, although possibly not enough to develop a reliable cognitive spatial map.

Components of Navigation and Locomotion

When assessing metaphors for navigation and locomotion in VEs, it is important to consider mappings of integral navigation and locomotion components to metaphor gestures or mechanisms. Of particular interest is the manner in which *direction* and *velocity* are performed as prescribed by candidate metaphors. For instance, in the walking-in-place metaphor, direction is determined by direction of torso and velocity by rate of walking — very natural mappings indeed.

[Templeman, 1996] classifies steering (directional) techniques into the following three categories:

- *body-based* — tracking torso orientation,
- *head-based* — tracking head position or gaze, and
- *hand-based* — tracking hand position or use of hand-held input devices.

Note that although the three classes were intended to describe steering or directional techniques, they may also be applied to velocity or propulsion techniques. For example, a hand-based technique for propulsion may require users to press a mouse button to move forward.

Body-based Techniques

Tracking a user's body or torso orientation is popular in body-based and lean-based navigation metaphors. Obviously, using body orientation is the most natural and intuitive means for users to specify direction. Moreover, ***body-based directional techniques allow users to perform related tasks concurrently*** ≪Nav11≫. For example, users may look in arbitrary directions while traveling, enabling users to perform many inspection and exploration based tasks easily. For example, when exploring new terrain and spaces, users may visually inspect environment features as they make their way toward a particular target. Another advantage of concurrent, related task performance via body-based techniques is that they free users' hands, allowing natural manipulations to be performed while navigating. Again, this interaction is most like that experienced in the real world, so users typically have little trouble learning and understanding how to interleave navigation and manipulation tasks.

As previously mentioned, the walking-in-place metaphor uses a body-based interaction technique to specify velocity; the rate at which users walk in place determines speed. A similar walking-in-place technique is under investigation at NRL. Of particular interest is the enhancement of user control during VE walk-throughs. Issues under investigation include understanding the distinctions among walking-in-place, stepping-to-turn, and turning while walking-in-place [Templeman, 1996]. Note that all three of these actions require very similar physical motions. Preliminary findings indicate that the task of turning while walking is easier under the walking-in-place metaphor than with hand controls.

Head-based Techniques

Head-based steering techniques use head orientation or eye gaze (indirectly) to determine direction. Since this technique forces users to look in the direction of travel, it does not allow them to "look around" while in motion. In some cases, however, ***head-based techniques are appropriate, such as when direction of gaze and travel are logically connected*** ≪Nav12≫. In these cases, this type of interaction may be beneficial, since making fine directional adjustments during navigation requires a simple shift in head direction. As with body-based techniques, users can use their hands for manipulation-based tasks [Templeman, 1996].

Various kinds of head-based navigation techniques have been developed for VEs; one that has had extensive evaluation for user performance is pre-screen projection, developed at the Naval Research Laboratory [Hix et al., 1995]. Pre-screen projection allows a user to pan and zoom integrally through a desktop VE simply by moving the head relative to the screen. The underlying concept is based on real-world visual perception, namely, the fact that a person's view changes as the head moves. Pre-screen projection tracks a user's head in three dimensions and alters the display on the screen relative to head position, giving a natural perspective effect in response. Specifically, projection of a virtual scene is calculated as if the scene were in front of the screen. As a result, the visible scene displayed on the physical screen expands (zooms) dramatically as a user moves nearer. This is analogous to the real world, where the nearer an object is, the more rapidly it visually expands as a person moves toward it. Further, with pre-screen projection, a user can navigate (pan and zoom) around a scene integrally, as one unified activity, rather than performing panning and zooming as separate tasks.

An empirical comparison of integral interaction techniques versus non- integral interaction techniques for panning and zooming in a desktop VE environment showed that integral techniques for integral pan and zoom yield better user task performance than non-integral techniques for those same tasks. Further, pre-screen projection positively influenced pan and zoom strategy adopted by participants when they used it before using a non-integral interaction technique [Hix et al., 1997].

Hand-based Techniques

Hand-based techniques such as tracking by glove, wand, joystick and so on are very popular in today's VEs, most likely due to ease of implementation. An advantage of hand-based techniques is that little or no physical "roaming" space is needed. Users can stand or sit in place while traveling arbitrary distances and directions. Thus, ***for VEs involving little or no manipulation, sitting users, or limited facility space*** (e.g., desktop and fishtank VE development and usage), ***tracking by hand may be more appropriate and comfortable*** ≪Nav13≫ for extended use.

Few means of using the head to specify velocity have been reported in current VE literature. An exception to this is the lean-metaphor which uses the distance between a user's head and waist to determine velocity [Fairchild et al., 1993]. Typically, head-based directional techniques are combined with hand-based techniques for specifying velocity. For example, MultiGen's SmartScene™ environment defines a rich set of two-handed interaction techniques including the ability to control velocity by varying the distance between tracked, gloved hands (see Figure 3).

Unfortunately, hand-based approaches to navigation are not well-suited for simultaneous manipulation-based tasks. Indeed, such tasks would be extremely difficult, if not convoluted, to design, much less use during navigation. Another potential problem with hand-based techniques is that in immersive HMD environments, users are not able to see the position or orientation of their hands or of their fingers relative to input device buttons and switches [Templeman, 1996]. While some users may become adept at this type of "blind" interaction, most users will likely experience increased task performance times and error rate, as well as disorientation and frustration.

FIGURE 3: MultiGen's Two-handed Interaction Technique [MultiGen, 1997]

Choosing intuitive, easy to use combinations of controls for specifying direction and velocity may be dependent upon application domain and relevant, appropriate metaphors. Results of a pilot study where users walk around in a virtual Naval ship [Templeman, 1996] indicate the following:

- for head-based steering, use *finger pressure* or *walking-in-place* for velocity control;
- for hand-based steering, use *finger pressure* for velocity control;
- for body-based steering, use *walking-in-place* for velocity control.

5.2.2 Selection of Objects

Selection of virtual objects is another basic user task frequently performed in VEs. As such, selection of objects is essentially integrated, at some level, into a large proportion of higher-level VE tasks. Indeed, a VE without the ability to select (and subsequently manipulate or query) virtual objects may have limited use. For example, a VE allowing architects to walk through a working design provides spatial perspective unavailable via traditional pencil and paper means. Now consider the same environment with selectable, manipulable, and queryable objects; this is a much more powerful tool available to the architect. The architect may now alter any number of objects within the environment — wall placement and shape, fixture locations and functionality, lighting design, etc. Load-bearing walls may be queried for wall composition and load information. The end result is additional functionality — effectively integrating previously disjointed tasks (e.g., calculating loads, altering design, evaluating effects of lighting, etc.) into a single, productive work environment.

How much task integration is good? Can VEs be overloaded?

We define the term "object selection" simply as the user acquiring control of an object or group of objects. Once acquired, any operation performed on an object or group of objects, such as relocating, reorienting, or querying, we term "object manipulation". Manipulation of objects is discussed in detail in section 5.2.3. In every day life, we select most objects simply by first obtaining proximity to the object (e.g., walk to the object) and then by orienting our bodies to make physical contact with the object. A possible problem with this description is that it fails to take into account other types of selection such as that involved in visual query. For example, we are able to gain a certain understanding of some objects in the real world by taking a good hard look at them. But have we acquired control of these objects? Thus, acquisition of objects may not be necessary in obtaining *some*, albeit limited, information about objects.

At a high level, developers of VEs should pose questions such as "what types of objects will users need to select?", "what potential spatial relationships exist between users and objects?", and "what types of interactions are appropriate given existing objects and spatial relationships?" In short, we must examine the extent to which selection mechanisms match user tasks and goals. Typically, selection mechanisms are derived from existing development environment features, code reuse, or legacy systems; these derivations may force users to *learn* non-intuitive interactions.

Selection of a Single Object

To gain a better understanding of object selection in VEs, we first consider the process of selecting one of n virtual objects. Selection may be based on a number of attributes including name, appearance, location, feel, etc. In turn, the attributes used may be an indicator of appropriate selection mechanisms. For example, ***selecting an object based on shape or spatial proximity may imply some form of direct manipulation*** ≪Select1≫. This is typical of everyday tasks; e.g., reaching for the mouse placed next to a keyboard or selecting a particular scalpel from a tray of surgical tools. On the other hand, selecting an object based on name or other non-spatial attribute may suggest a form of indirect manipulation such as verbal query.

Selecting objects from a distance introduces other issues. As previously mentioned, selecting objects at a distance in the real world typically involves first positioning ourselves in proximity to the object. However, in VEs we are able to break real-world limitations and restrictions, opening doorways for new and powerful interaction techniques.

When selection of objects at a distance is done visually, some spatial concerns arise. In particular, the greater the distance between user and object, the smaller the object appears to that user [Gibson, 1986]. Given the technology of displays, some objects may be reduced to single pixels. ***Using non-linear scaling or minimal object sizes may help users discern one distant object from another*** ≪Select2≫. Tightly clustered distant objects pose further visual and spatial selection challenges, suggesting that indirect manipulation methods may be more appropriate.

Selection of Multiple Objects

Some tasks require higher level interaction and conceptual analysis. In these cases, ***selection of multiple objects may be extremely useful*** ≪Select3≫ [Templeman, 1997b]. One method of accomplishing selection of multiple objects is to individually name each member. This naming can be done via direct manipulation, where a moded command or interaction signifies building of a set as opposed to single object selection. Selection may also be based on membership in other sets. That is, users may wish to select all instances of some meta-object such as "all enemy tanks."

For selecting multiple objects based on spatial relationships, ***bounding boxes***, ***marquees***, and ***"rubber bands"*** ≪Select4≫ are useful. Much like two-dimensional desktop selection, bounding boxes allow fast localized grouping. This type of selection mechanism is typically performed within a user's working volume or within arm's reach [Mapes and Moshell, 1995]. As such, selecting spatially collocated objects via a bounding box does not work well for long-range selection. Since this selection technique is solely based on location and spatial arrangements, it does not work for selection tasks based on relations. For example, a bounding box will not be able to "select all enemy tanks within a five mile radius of headquarters."

For more powerful and flexible selection solutions, selection may be based on special criteria — via a query-based selection technique. Queries may be based on attributes including straightforward characteristics of objects, temporal relations, and spatial relations. For example, in a command and control application, users may wish to select "all enemy tanks which have moved to within a 20 mile radius of headquarters in the last 24 hours." Once selected, further inquiry may be performed on the set as a whole, or on single members within the set. In general, ***when selection criteria are temporal, descriptive, or relational, non-direct manipulation selection techniques*** (e.g., selection via query) ***are appropriate*** ≪Select5≫.

Selection Feedback

Whether selecting one or multiple objects, ***providing users with some form of selection feedback is paramount*** ≪Select6≫. Without some form of feedback, users will not be able to ascertain which, if any, objects are selected. Given the highly graphical nature of most VEs, indication of selection is typically done visually. Some examples of visual indication include highlighting the objects using distinct coloring, and outlining of objects by bounding them with a cube or other polygon. When assessing visual indication, it may be useful to consider spatial relations between users and objects. For example, small objects may need large bounding polygons to facilitate selection from a distance. Simply outlining an object which is small and distant may not be sufficient. On the other hand, a bounding polygon which is too large may overlap nearby objects, introducing ambiguity into the selection process.

Environments which rely heavily on haptic information may benefit from force-feedback or tactile indication. For example, in remote sensing environments, operators typically use video monitors for large scale movement and orientation, but depend on haptic feedback for more detailed tasks such as grasping and manipulation. In such cases, a video feed may not provide appropriate views or details required for successful and efficient task completion.

Aural feedback, such as voice confirmation or continuous tones, is another option. Typically these types of indication are better used redundantly, as reinforcement of visual indication. For example, a voice confirmation alone provides limited persistence so that users may not be able to identify which objects have been selected seconds after selection. On the other hand, when coupled with visual indication, voice confirmation may aid in distinguishing closely placed, small, or distant objects from one another. This technique may integrate well into voice-driven selection mechanisms. In general, it is not desirable to mix interaction modes and senses. Thus, if a user selects objects by voice, they may expect to also receive indication and verification of selection via aural channels.

Another important consideration in design of selection mechanisms is use of visual transparency; that is, graphically rendering user avatars, cursors, and other pointing mechanisms semi-transparently. ***Use of transparency during selection helps avoid occlusion and hence yields a better working context*** ≪Select7≫ [Hinckley et al., 1994a]. As Hinkley points out, studies have shown transparency to be just as, if not more, useful for dynamic target acquisition than stereopsis [Zhai et al., 1994].

Types of Object Selection Methods

One class of selection methods is based on the notion that users directly manipulate objects in the VE. Well represented in HCI literature, direct manipulation has been shown to increase usability in traditional user interfaces [Shneiderman, 1992] [Hutchins and Norman, 1986]. Given the interactive goals and tasks of most VEs, it no surprise that direct manipulation interaction techniques are very popular in VEs.

Target Acquisition

A common direct manipulation selection technique which relies heavily on three-dimensional collision is dynamic target acquisition. Using this technique, users must position a user avatar, cursor, crosshair, or other pointing icon in three-dimensional space so that objects of interest are intersected. This type of selection task in non-trivial since humans do not innately understand three-dimensional space, but instead experience three-dimensional space [Hinckley et al., 1994a]. Subsequently, complex three-dimensional "docking" tasks require good spatial aptitudes and hand-eye coordination.

Fortunately, techniques exist to help users better understand and perform dynamic target acquisition in three-dimensional space. VE input and output devices should ***strive for minimal lag as well as a certain level of display and device fidelity*** ≪Select8≫ — two traditionally opposing characteristics. On one hand, studies on lag in VEs have shown that the feedback loop between user input device and visual display affects target selection [Ware and Balakrishnan, 1994]. On the other hand, high fidelity devices afford precise positioning and registration in three-dimensional space. For example, a 1996 study shows that high frame rates and stereoscopic imagery may increase task performance in dynamic target acquisition tasks [Richard et al., 1996]. Subjects in a desktop or fishtank VE were instructed to grab a moving red ball out of the "air" using a data glove equipped with force-feedback. Results showed that selection time was decreased when the haptic interface was coupled with stereoscopic viewing and high frame rates. As [Richard et al., 1996] points out, the existence of haptic feedback, high frame rates, and stereoscopic viewing require less adaptation on the part of users since these conditions are closer to natural, real-world interaction.

Typically users are required to first intersect an object of interest with the cursor. Object ***intersection points should be made as obvious and accessible as possible*** ≪Select9≫, with very small objects potentially having relatively large intersection areas. Once intersected, users may, for example, press a button, make a gesture, or speak a command to indicate that the intersected object is to be selected. A problem with this approach is that users normally have trouble keeping cursors perfectly still, so that pressing a button will move the cursor, effectively de-selecting the object. ***To aid in efficient selection of objects, damping, snapping, and/or trolling may be used*** ≪Select10≫.

Damping may be applied when a user's pointing icon approaches a potential target, providing additional visual stability for the cursor. Snapping forces a cursor to jump to a nearby object. Both these techniques may be useful when objects are small and placed within a small area. Trolling or fishing is a useful technique for increasing user performance in three-dimensional target acquisition tasks. As users move an avatar or cursor through space, intersected objects "stick" or "hook" onto the cursor. This type of object selection requires a bit less coordination than the alternative (i.e., position cursor then select) method. A problem with trolling is that undesirable objects can be hooked en route to the object of interest. Once hooked, the undesirable object must be released, and without some type of undo command, will have been effectively moved to some new, possibly undesirable, position.

[Hinckley et al., 1994a] suggests that ***targeting of three-dimensional objects should be based upon relative motion*** ≪Select11≫; that is, grounding a user's relative coordinate system with a spatially familiar object such as the user's body, the user's other hand (as used in two-handed interaction — see section 5.2.3), a real object such as a desktop, or the starting point or origin of gesture.

An excellent example of targeting three-dimensional objects relative to the user's body is Ivan Poupyrev's go-go interaction technique [Poupyrev et al., 1996]. By using a non-linear mapping of the user's physical hand to the virtual hand, a user's physical working volume may be expanded to fit the size of larger virtual spaces. That is, when the physical hand is close to the body, the virtual hand is close to the virtual body. However, at arm's length, the virtual hand may be mapped to arbitrarily large distances, allowing a user to select objects at great distance simply by reaching for them. One of the most obvious benefits of this technique is that it is extremely intuitive — users reach for objects just as they would in the real world. Another benefit is that the technique facilitates seamless interaction among objects at a distance and objects close at hand. This cannot be said of non-relative, or absolute, acquisition designs, where positioning a cursor on distant objects is tedious and tiresome. One disadvantage may be that users find this technique confusing, at least initially, since arms in the real world do not have unlimited reaching capability.

Ray and Cone Casting

Two other closely related types of direct manipulation selection techniques are that of ray casting and cone casting. These techniques project a ray or cone, respectively, from a user's hand into the VE. More often than not, the ray or cone is projected at such an angle that users need only point at objects of interest. That is, the technique employs an extended pointing metaphor.

Ray casting should be used when objects to be selected are very small or collocated among other objects ≪Select12≫ [Hinckley et al., 1994a]. Likewise, ray casting works well when selecting a small object placed among many others. An important consideration in using ray casting as a selection method is specifying what parts of the ray are valid intersection points. Some designs use the terminus, or tip of the ray, for intersection. However, there are problems with this approach. To facilitate distant object selection, users must either dynamically alter the length of the ray, or the system must dynamically define the ray's terminus based on intersection of predominant planes and surfaces within the VE. In many cases, objects and points of interest do lie on some surface, so ray casting can be very useful. For example, a weather station visualization tool developed at the National Center for Supercomputing Applications (NCSA) [National Center for Supercomputing Applications, 1996] uses ray casting to select weather stations positioned along the surface representing a large geographic area. By fixing the ray angle parallel to the y-axis (up and down) and fixing the geographic surface parallel to the x-axis, selection of stations along the surface is simplified. In this case, the nature of the application facilitates restriction of ray and surface angles resulting in an intuitive, easy to use interface.

Under what conditions does restricting users and/or objects become useful?

Cone casting is useful in applications where selection of regions or of large, sparse objects is required. Like ray casting, designers must consider which part(s) of the cone are valid intersection points. ***"Spotlighting", or highlighting the base of the projected cone, may be useful as it provides visual spatial cues for intersection*** ≪Select13≫. [Liang and Green, 1994] describe some designs and more detailed uses of spotlighting in object selection. Selection of a distant object with projected cones is difficult since the base of such cones increases in size with distance to desired objects. In these cases, providing some mechanism for users to adjust cone widths may be advantageous. Indeed, an infinitely small cone width is essentially a ray, thus providing users with the best of both cone and ray techniques.

While object selection via ray and cone casting appears to be useful for some applications, there are some usability concerns which should be addressed. For instance, both rays and cones suffer from occlusion problems; that is, the ray or cone may occlude background objects of interest. As previously mentioned, the use of visual transparency decreases the occlusion problem to some degree. Another problem is the fact that a ray or cone can project through many objects, if objects are in line with respect to viewing angle. In these cases, a means to toggle among in-line objects must be designed, resulting in a more complicated selection technique. Another potential problem arises from the "fulcrum effect" where small movements close to the body may large-scale movements at the tip of the ray or cone. Lastly, and perhaps most significantly, is the fact that both ray and cone selection are merely ways to select objects; they do not facilitate seamless object selection *and* manipulation. Again, a separate method is needed to actually manipulate the selected object(s).

5.2.3 Object Manipulation, Modification, and Query

VE users typically select objects of interest with the intent of performing some type of manipulation, modification, or query on that object. The types of actions available to users for given objects, in part, defines the power and usefulness of the environment as a whole. For example, a VE containing objects of limited functionality may allow users to merely look at or grasp these objects. While this conveys some useful information about these objects, such as size, location, orientation, and so on, it does not allow for more insightful inquiry and use such as related database entries, user annotations, and inherent functionality. For example, the ability to query real-time databases associated with a virtual ship along a shoreline supports a more useful and powerful VE than a VE which does not support object query and complex manipulation.

We define the term "object manipulation" to be any actions performed on virtual objects once selected. These actions may range from simply relocating or rotating objects to changing object attributes and behavior to performing complex object queries. [Kijima and Hirose, 1996] states that "manipulation is realized by calculating the behavior of the object driven by a virtual hand." While this definition places appropriate emphasis on the behavioral nature of user-object interaction, it does not take into account indirect means of manipulation such as speech recognition, command menus, and gesturing.

Important for Manipulation

Regardless of manipulation means, users must be given enough relevant and accurate information to perform manipulations easily. [Wickens and Baker, 1995] identifies the following issues as important for object manipulation:

- *accurate* depiction of location and orientation of surfaces,
- *minimum* display lag,
- *multimodal* interaction, and
- *spatially relevant* and *revealing* user point of view.

[Wickens and Baker, 1995] points out that ***accurate depiction of location and orientation of surfaces*** ≪Manip1≫ yield critical visual cues used to judge motion and orientation. Related work emphasizes the importance of "regularly spaced texture, and level surfaces to the accurate perception of gradients, slant, and the time until a moving object collides with a solid surface."

Since most manipulations require high spatial and temporal precision, ***minimum display lag*** ≪Manip2≫ is necessary. Large time delays may cause users to over- or undershoot targets, resulting in a frustrating experience. For example, [Sturman et al., 1989] showed that a display running at six to ten frames per second was much more natural than the "impossible to use" display running at only three frames per second.

Multimodal interaction ≪Manip3≫ reinforces cognitive perception during manipulation tasks by providing alternate sensory input. For example, studies have shown that the addition of aural, haptic, and force feedback cues to otherwise visual-only systems improve user perception, manipulation, and performance [Brooks et al., 1990].

A ***spatially relevant and revealing frame of reference*** ≪Manip4≫ or point of view is another issue important for manipulation tasks. Obviously, users must have a clear, unobstructed, detailed view of the manipulated object(s). Moreover, depending upon the type of manipulation task, users may need an egocentric, exocentric, or simultaneous views.

Gesturing to Manipulate Objects

Many VEs use some form of hand gesturing to facilitate object manipulation. Gesture interaction is natural for most users, since humans frequently gesture and work with their hands. Gesturing is supported by a variety of data gloves, most of which are able to distinguish finger movements down to the joint level, and thus are able to support a large number of gestures. More detailed discussion of data gloves as input devices is given in Section 7.2.3. As a general rule, ***VEs should avoid non-intuitive, unnatural, or poorly-mapped gestures*** ≪Manip5≫.

One the more significant advantages of gesturing is the ease of association between gesture and meaning. Indeed, many gestures used in the real world have clear, direct meaning in a virtual world. For example, a *pinch* metaphor is typically used for picking virtual objects. As in the real world, users pinch virtual objects between their forefinger and thumb to grab them. Some interfaces attach different meanings to different forms of pinching. For example, a pinch with the index finger and thumb may signify object selection, the middle finger and thumb may signify stretching of an object, the ring finger and thumb may signify object rotation. From a development point of view, implementing this type of interaction is trivial given a pair of pinch gloves. However, from a user's point of view, this type of interaction may be hard to learn and remember, since the pinch gesture is effectively overloaded. Users will typically remember "what" can be done, but may have a hard time recalling which fingers to use. ***When defining moded pinch commands and mappings, developers should keep in mind user experience levels*** ≪Manip6≫.

A closed hand, or fist, gesture may be used to denote *grasping*. Users reach for objects, then perform a grasp gesture to indicate acquisition of an object. To retain control of an object, users have to keep their hand in a closed position. This may limit the types of other manipulations available to users, such as large-angle rotation or scaling. To release control of objects, users simply open their fists.

A problem inherent with implementing gesturing as a means of object manipulation is the fact that natural gesturing involves a series of transitions from gesture to gesture, essentially creating a continuum of gesturing [Mapes and Moshell, 1995]. This makes recognizing gestures very difficult, as the hand and fingers may be constantly moving. Moreover, gestures may differ slightly between users and even during a session. In order to guarantee recognition, gestures may have to be exaggerated, creating uncomfortable positions for users [Mapes and Moshell, 1995].

Two-handed Interaction and Gesturing

[Hauptmann, 1989] asserted that systems restricting users to one-handed interaction will be inadequate for most commonly used manipulations. Instead, many VE interface designers are realizing that ***two-handed techniques provide more powerful and natural gesture interaction*** ≪Manip7≫. Moreover, the use of two hands allows for more complex, intuitive manipulations such as rotating, scaling, and translating [Mapes and Moshell, 1995]. Many of the tasks undertaken in the real world are performed, almost subconsciously, using two hands. Subsequently, VE users are likely to attempt two-handed interaction, even when such interaction is not supported. [Hauptmann, 1989] observed users spontaneously using two hands to perform object rotation, translation, and scaling.

Two-handed interaction plays an important part in supporting user perception of three-dimensional space. Allowing the use of both hands helps users ground themselves in the interaction space [Hinckley et al., 1994a]. Moreover, additional spatial grounding may be achieved by exploiting peoples' natural ability to understand the position of their hands relative to each other and to their body. Informal observations made at the University of Virginia showed that users interacting with two hands are less likely to become disorientated than users interacting with one hand [Hinckley et al., 1994a].

Related work in the field of motor behavior provides useful insight into how humans use both hands to perform tasks. In particular, it has been shown that the ***non-dominant hand is generally used for large-scale, coarse positioning while the dominant, or preferred, hand is used for fine-grained tasks*** ≪Manip8≫. Moreover, humans position their dominant hand relative to the coordinate system specified by the non-dominant hand [Guiard, 1987]. For example, consider the simple task of writing a letter. The non-dominant hand usually holds a piece of paper in place, while the dominant hand performs detailed movements with respect the orientation of the non-dominant hand.

The implications for design based on these findings include assignment of devices to support tasks as well as the assignment of devices to user hands. VE developers wishing to implement two-handed interaction should first identify representative tasks and then consider how each of these tasks would be performed using two hands. In particular, developers need to analyze how two-handed performance of each task is accomplished relative to the dominant and non-dominant hands. Once tasks are considered at this level, the second implication for design is quite trivial, namely assignment of device to hand. Assuming the device(s) will fit comfortably in either hand, the assignment *should* be as simple as asking users "are you right-handed or left-handed?" Unfortunately, flight sticks, joysticks, and, in particular, user-worn gloves usually favor the right hand.

However, designing the details and specifics of an intuitive two-handed interface is not a trivial task. For example, in a simple object manipulation study, researchers observed that users typically used two hands for orienting an object (rotation), and used only one hand for object translation [Mapes and Moshell, 1995]. During single-hand translations, slight perturbations caused the orientation of the object to change (slightly rotating one direction or another), forcing users to correct the orientation after translation. These observations led to a redesign, where one-handed interactions simply translate, and cannot rotate. This type of restriction matched users' natural task performance method, so that the restriction was not a hindrance in the end.

Object Rotation

A fundamental tasks in many VEs is that of visual, aural, or haptic inspection. Much like real-world experiences, VE users typically grasp objects of interest, move them closer, and perform a more detailed inspection. Critical to the process of inspection is *object rotation*.

One of the more fundamental problems with object rotation is the natural restrictions and limitations of human wrist design. That is, using only a grasp, the available rotations of the wrist do not afford all possible viewing angles. Some more complex object rotations are obtained in the real world by combining wrist rotation, finger manipulations, and detailed haptic feedback — a suite of interactions not fully integrated and matured in today's VEs.

Clutching and Ratcheting

A common solution to this problem is to ***incorporate some type of clutching mechanism*** ≪Manip9≫ into the interaction technique. Clutching allows users to reposition or reorient input device(s) without effecting VE system state. That is, the relationship between a user's *physical* position and a corresponding *logical* or *virtual* position may be altered so that users may specify a comfortable resting position. Clutching is crucial to task performance involving large degrees of rotation. Caution should be taken, however, as a poor clutching mechanism can jeopardize the usefulness of spatial input and tasks

[Hinckley et al., 1994a].

Some researchers have implemented clutching mechanisms as a *clutch button* mounted on a spatial input device. While this may seem like an appropriate location for such a button, there are some task-related problems with placing a clutch button in a fixed location. In particular, if representative tasks involve arbitrary, large-angle rotations, then a fixed location of the clutch button implies a fixed grip on the input device. This in turn restricts user movements, and can be awkward during rotations [Hinckley et al., 1994a]. In these cases, separating the clutching mechanism from the primary spatial input device may be a better design. For instance, a voice recognition system developed at SRI International allows user to initiate clutching with a simple voice command [SRI, 1996].

While clutching, users are free to orient the spatial input device to any position. On the other hand, if user tasks do not require these types of rotations, then fixing the location of a clutching button may be more appropriate. Furthermore, mounting the button on the input device it controls suggests something about its functionality. That is, users are more likely to assume that a button on a spatial input device will affect its operation than a button on another device [Hinckley et al., 1994a].

Other solutions to the problems associated with object rotation include attaching virtual levers to objects [UVA, 1996]. Rotating the object is then done by grasping the end of the lever and moving it in a motion around the object. Another solution is to have virtual objects "float" in front of users when under inspection, essentially replicating a gravity-free environment. To completely rotate an object, a user essentially "ratchets" the object using a sequence of small rotations. Between rotations the object remains still.

Object Modification

Aside from spatial manipulations such as rotation and re-location, VEs may also support object modifications, or altering of object attributes. For instance, a VE designed to support architectural or interior design may need to provide the ability to dynamically resize, scale, or deform various objects such as walls and furniture. This type of interaction supports more productive work as compared to the traditional "alter source code then re-compile" scenario.

Allowing users to alter basic object attributes, such as color, shape, labels, and so on, ***is another useful feature*** ≪Manip10≫. Customizing objects within a VE allows users to organize the virtual space in a manner meaningful to them. In turn, users develop and retain a more elaborate cognitive map of the virtual space. In addition to altering object attributes, users may wish to alter an object's behavior to serve specific, immediate needs. For example, a VE designed to train military users may support a wide range of enemy behavior and engagements by allowing users to specify the behavior of enemy forces. Likewise, a VE designed to support dynamic visualization of chemical reactions may support user manipulation of atom or element behavior.

Querying of Objects

Generating intuitive, powerful methods for performing queries and displaying query results in VEs is a hard problem, and to date, remains an open research question. We will consider two general uses of queries in VEs: as a form of selection and as a means to retrieve information. Queries which result in selection we term "selection queries." ***Combining query formation with selection methods*** ≪Manip11≫ (see Section 5.2.2) is hard but powerful [Esposito, 1996]. In such cases, users typically wish to select objects based on non-spatial, context-sensitive, or descriptive characteristics. For example, users onboard a virtual warship may wish to select all neighboring ships that have suffered significant damage. Once selected, users may continue with other tasks relevant to the selected group of objects. Selection queries should result in the selection of one or more objects or some user notification that no objects meet the query criteria. Objects that do meet query criteria should be selected, with indication of selection (e.g., highlighting) consistent with other forms of selection.

How can VEs support complex query formation and meaningful presentation of query results?

Queries which result in the disbursement of specific information we term "informative queries." One of the simplest types of informative queries are those which return general information about a particular object. This type of query can easily be done though direct manipulation. For example, SeaDragon, a command and control application developed at the Naval Research Lab (NRL) for the Marines allows users to query elevation, latitudinal, and longitudinal information from any point in the terrain and air space by first positioning the cursor followed by pushing a button [Naval Research Laboratory, 1997]. The same application allows users to query tank and aircraft objects in a consistent manner, displaying information such as allegiance (friend or foe), type of vehicle, and position in latitude and longitude.

A richer type of informative query is that which allows users to query one or more databases through virtual objects or non-direct manipulation means. For example, SeaDragon may serve as a visualization tool of real-time intelligence data streaming in from many database sources. In this case, the types of queries available to users are much more dynamic, powerful, and informative than general queries. The wealth of information contained in such databases coupled with simulation allows users to perform "what if" and probabilistic types of queries. Extending SeaDragon, users may perform queries such as "display extent of possible troop and tank movements over the next 24 hours given that rain and fog are likely." This query would rely not only on updated database information for positioning of enemy forces, but also terrain and weather data.

A different type of informative queries in VEs are ***interface queries***. These queries ***allow users to determine what types of actions are available for given objects*** ≪Manip12≫. [Esposito, 1996] gives some examples of interface queries, such as:

- what can I *do* to this object,
- does the object have any *behaviors*, and
- how do I *invoke* these behaviors?

Developing methods for presenting the results of user queries in a useful and unobtrusive manner may be as hard a problem as developing methods to perform queries. Available means of presentation include audio, textual, and graphical channels. In each case, considerations for data sensitivity and user privacy should be honored, integrating private displays or headsets as necessary [Esposito, 1996]. The manner in which query results are presented depends upon both the application as well as the nature and form of the query. For example, as mentioned above, when a query is used for selecting objects, a graphical means consistent with other available selection mechanisms should be used (e.g., highlighting). When queries seek information, rather than selection or control, then a textual representation may be more appropriate. Displaying textual information in three-dimensional VE space is yet another area worthy of investigative research.

5.3 Tasks, Task Sets, and Application Areas

Autonomous analysis of typical VE user tasks, such as navigation, manipulation of objects, and querying objects paint only part of the usability picture. For a more complete understanding, it is useful to consider the implications of combining tasks, in particular, how lower level tasks combine to form higher level tasks.

5.3.1 Task Sets

The most basic user tasks, which we term "primitive", form a base from which more complex tasks are built. An example of a primitive task may be one of simple movement, such as the movement of a finger. By definition, primitive tasks are the most simple tasks. They are also the most general in terms of use. Indeed, primitive tasks are found in *every* high-level task.

A set of primitive tasks we term a "composite" task. For example, a user performing the primitive task, *move finger*, using all four fingers and thumb in the correct order, is essentially *grasp*ing. Combining the tasks *grasp* and *move arm* results in a high-level task, *acquire object*.

A set of tasks, or *task set*, is a grouping of several tasks, with a defined relationship among them. The union of the member tasks should be meaningful and result in some common, greater activity. Performance of all member tasks (in some hierarchical order) represents performance of a single high-level task.

The hierarchy of low-level tasks during performance of a high-level task may be dependent on the nature of the high-level task as well as relationships among low-level tasks. That is, some of the low-level tasks may be done in parallel, while others may need to be performed serially. For example, to *acquire* an object, the user must *move* their arm to position their hand above the object **before** the user *grasps*. A more thorough discussion of the temporal aspects of user tasks is given in Section 5.1.3. As the complexity of the task set increases, the applicability of the task becomes more specialized. Indeed, a collection of complex tasks which make up brain surgery is incredibly specialized. Figure 4 graphically depicts a possible relationship between task complexity and task applicability.

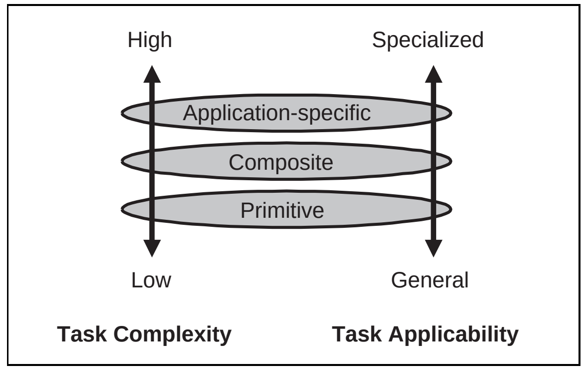

FIGURE 4: Task Complexity vs. Task Applicability

A specialized task such as brain surgery we term an "application-specific" task — a collection of application-specific tasks characterize a particular *application area*. For example, brain surgery is one of many application-specific tasks within the application area of *medical simulation*. Figure 5 shows the hierarchical nature of primitive, composite, and application-specific tasks within an application area.

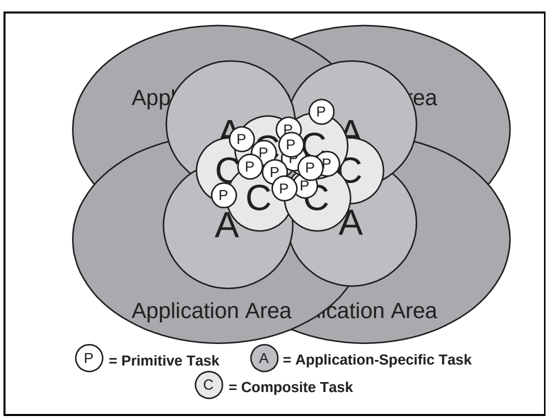

FIGURE 5: A Possible Relationship Between Application Areas and Task Sets

Understanding the relationship between an application area and its composite tasks may help VE designers identify potential user goals and tasks. Knowing the application area(s) in which a given application belongs exposes a set of high-level application-specific tasks normally associated with the application area(s). Recall that each of these application-specific tasks is composed of several lower-level composite tasks which are in turn composed of several primitive tasks. By enumerating the lower-level tasks within an application-specific task, a designer can make use of previous designs and evaluations of such lower-level tasks. Furthermore, any usability concerns associated with these tasks are brought to light for consideration, providing additional information for use in design (or assessment) decisions (e.g., avoiding common pitfalls, or "reinventing the wheel"). In sum, it appears that a focus on user tasks is as important in VEs as it is in traditional 2D computer applications.

5.3.2 Application Areas of VEs

Current research and development in VEs appear to be clustering into a handful of application areas. As in any other emerging field, early VE researchers and developers sought a meaningful, useful context in which to utilize the new technology. But do current application areas represent such "a useful context"? Is it possible that developers and researchers are haphazardly throwing everything into the VE ring, without knowing whether VE is appropriate? As a community, VE researchers and developers should aim to identify and investigate application areas for which VE technology is well-suited. Figure 6 lists a few of the more common current application areas of VEs.

Medical training and simulation
Military training and simulation
Decision support / Command and control
Scientific Visualization
Engineering design
Architectural design
Entertainment
Teaching and Education Manufacturing

Figure 6: Some Common Application Areas of VEs

Some usability characteristics are more critical in one application area than another. For example, precision, accuracy, and lag associated with registration are of greater concern in a surgical simulation than in an entertainment setting. Usability of an entertainment-based VE may rely more on characteristics such as the fidelity of audio, setting, etc.

What usability characteristics are of greatest concern for a given application area?

6 The Virtual Model

Consider the vast amount of naturally occurring information we are able to perceive via our senses. As living creatures, we instinctively use this information, interpreting it to create a mental picture, or model, of the world around us. Users of VEs rely on system-generated information, along with other information, such as past experience, to shape their cognitive models. Users also interact within such system-generated information spaces, so that the information flow is essentially bi-directional. We term the abstract, device-independent body of information and interaction the "virtual model." The virtual model defines all information that users perceive, interpret, interact with, alter, and — most importantly — work in.

6.1 Characteristics of Virtual Models

The meaning and relevance of presented information are important considerations when assessing the usefulness of presented information. In general, both the semantics and presentation of information in VEs can be viewed as:

- *clear* or *abstract*,
- *simple* or *complex*,
- *relevant* or *ornamental*, and
- *consistent* or *specialized*.

In general, clear, simple, relevant, and consistent information obviously is desired, but these criteria are not always attainable or even desirable. For instance, an abstract and ornamental world is very appropriate for a VE designed for meditation purposes. Thus, the semantic characteristics given above represent a *continuum* of characterization. Most VEs will "fit" into the continuum in a unique way. Understanding the relationship between a VE purpose and its "fit" may help developers identify usability weaknesses and strengths, with the potential to suggest a solution (e.g., Midshipmen find the database query mechanisms too complex for real-time, wartime use, and suggest a simpler interface).

Other characteristics of information stem from the fidelity of representation. These characteristics, which expose the degree to which the representation accurately portrays an entity, include whether the representation is, for example,

- *rich* or *minimal*,
- *true to real-world physics* or *abstract*, and
- *dynamic* or *static*.

Some imagine VE as potentially being as dynamic, rich, and true-to-life as the real world. Research is revealing that such a high-fidelity replication of the real world is not only technically challenging, but possibly not always useful. Simple, informative VEs that actually facilitate real work may provide much more utility than high fidelity, true-to-life VEs. These characteristics are discussed as they apply to classes of information presented (e.g., users, agents, system information) in the following sections.

6.2 Types of Information Present in Virtual Models

Figure 7 lists four types of information typically present in virtual models, along with instances of each. Note that this typing scheme is not intended to be a rigorous classification. Each type represents a class of information with common characteristics.

"VE users" and "VE agents" introduce different sets of usability issues into VEs, so it is useful to examine each separately. The line between agent and user may be blurred in analyzing situations with multiple users. Here, whether or not a given user is considered to be an agent depends upon the point of view; I am certainly the user from **my** point of view. However, another user may benefit by viewing me as an agent. [Trias et al., 1996] offer an insightful distinction between users (avatars) and agents: "An avatar is a virtual agent controlled by a *user*, the *user* provides both the decision making and the motion behaviors, while the avatar mimics these movements into animation."

Type	**Type Instance**
VE Users	Disembodied hand avatar, cursor
VE Agents	Virtual dog, “talking head,” virtual servant
Virtual Surrounding and Setting	Ship, house, Ft. Benning
System Information	Command windows, application-specific Information

FIGURE 7: Types of Information Present in Virtual Models

The “virtual surrounding and setting” consists of all objects, other than users and agents, which make up the environment’s setting (i.e., battlefield, ships, landscape, walls, furniture, horizon). “System information” includes items such as the command interface, application-specific data, and online help.

How a type instance is presented has the potential not only to improve the usability of a system, but to hinder the usability of a system. Explicitly addressing usability characteristics within a type helps to determine how a type instance may affect the usability of a system. Some usability characteristics for each type of VE information are discussed below.

6.2.1 VE User Representation and Presentation

The presence of our body has a profound effect on essentially everything we do. Indeed, it provides immediate and continuous information about our presence, activity, attention, location, identity, availability, status, and much more [Benford et al., 1995]. Consider the task of representing the user *within* a system, a common occurrence in VEs. Unlike traditional 2D interfaces, usability is affected by how the user is presented, both to oneself as well as others.

Semantics and Relevancy of the User's Representation in a VE

The semantics of representing the user within a VE may pose the following questions: *What does a user's avatar mean?* and *Is the user's representation even relevant to the task at hand?* Current literature suggests that the presentation of a user serves many purposes depending upon the audience. For example, to other users, ***presentation of a user conveys activity and viewpoint*** ≪UserRep1≫. To a user, ***presentation of that user increases presence by providing a familiar frame of reference*** ≪UserRep2≫ [Benford et al., 1995].

Point of view or a *viewpoint* represents the area in space that the user is attending. Visually, we usually consider viewpoint to be the volume of space viewable given field of view, occlusion, and other constraints. A viewpoint frames the user's attention to other senses as well, such as the direction and space where the user is listening [Benford et al., 1995]. Viewpoints are generally considered to be either *egocentric*, referring to a viewpoint from within the user, or *exocentric*, referring to a viewpoint from outside the user. An ***egocentric point of view is useful when users need to experience a strong sense of presence*** ≪UserRep3≫. ***Situations where users benefit from an exocentric view include ones in which detailed relative position and motion between the user and other objects*** ≪UserRep4≫ are desired, e.g., orienting oneself in a large environment by externally viewing oneself in the context of the environment.

The spaces attended by a user's senses are not necessarily always the same. Furthermore, these spaces may or may not correspond to spaces in which a separate task is being performed (e.g., object manipulation). For example, when driving a car, the user may be *manipulating* the accelerator, *looking* straight ahead, and *listening* to the passenger next to them. This example illustrates another important consideration: the spaces attended to by multiple users may or may not coincide, either by accident or design, and often depend on each user's current task goal.

User-Specified Representation and Presentation

Because a user (and others) attends to viewpoints from many senses, the representation of that user within the system may need to provide appropriate presentation at each of the sensory levels. That is, a user can be represented not only visually, but aurally, tactilely, etc. The degree to which presentation accurately reproduces the real world is termed "fidelity." The fidelity of an entity's presentation in a VE can be examined as a whole and at each sensory level.

Representing the user within a VE is known as *user embodiment*. A high-fidelity embodiment supports life-like presentation of the user at every sensory level. A low-fidelity embodiment is the simplest of user representations, such as a graphical block, and may only support a single sensory level (e.g., visual).

User embodiments should be as efficient as possible ≪UserRep5≫ [Benford et al., 1995]. That is, the embodiment should only include useful and relevant content, detail, and sensory representation. Benford suggests that attempting to reproduce the physical human form in full detail may be wasteful. The additional computing power required for such a reproduction may outweigh any benefits obtained from such a reproduction. Benford further suggests that more abstract forms of embodiment may be more appropriate. The abstract form should be as simple as possible, yet still convey relevant, useful information (presence, location, etc.). Benford bases this argument on the inevitable lack of computing resources. This then leaves the question: *Assuming that sufficient computing power exists, do we want embodiments to resemble our real-world physical form in as much detail as possible*?

How much should embodiments resemble users' real-world form?

One answer to this question may be to ***allow users to control presentation of both themselves and others*** ≪UserRep6≫. For example, in an information-rich environment, a user (or agent) may be portrayal on many different levels (e.g., life-like appearance, text-only information, audio, application-specific information). The ability to specify all or some of these levels affords the benefits of customization. Furthermore, by supporting the representation of users on many levels, a system is in fact supporting graceful degradation. For example, users in a networked environment may not all be using the same kind of hardware. By supporting graceful degradation, the user on a less powerful machine uses a less intensive, lower-fidelity presentation without affecting the presentation to those on more powerful machines.

Supporting the ability to alter point of view, or viewpoint ≪UserRep7≫, also appears to be worthwhile in VE systems. For example, [Stoakley et al., 1995] suggests that the ability to view scenes and objects from many different angles allows users to gain a better understanding of the environment.

Examples of User Representation

SRI International — A networked, distributed collaboration tool developed at SRI International uses a 2D picture frame to represent the user, labeled with user-specified text. A static 2D image can be placed within the frame for a more detailed representation. Though simple, this representation is adequate for conveying critical exocentric, spatial information [SRI, 1996].

University of Virginia, Charlottesville, VA — In the World in Miniature (WIM) model, a camera icon is used to represent the user's position and orientation. The user can change location by simply grabbing and placing the camera icon in a new location. Another avatar design in the WIM uses a virtual doll to represent the user. The doll's head moves in response to the user's tracked head. A view frustum can be added so that the doll's field of view is explicitly shown. A frustum is the portion of a pyramid formed by cutting off the top via a plane parallel to the base. In the WIM, the frustum is a four-sided pyramid representing the user's viewpoint (see Figure 8). The

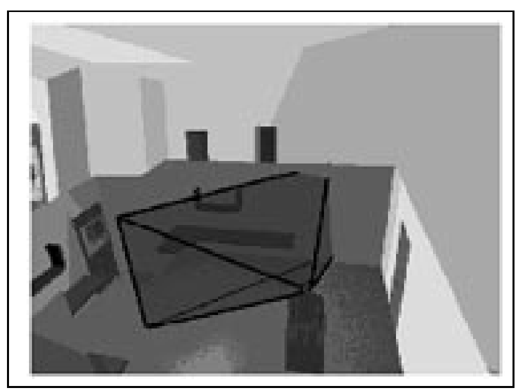

FIGURE 8: User's Viewing Frustum within [Stoakley et al., 1995]'s WIM

frustum extends from the doll's eyes out into the WIM and helps users understand the spatial relationship between their current field of view and the surrounding environment [UVA, 1996] [Pausch et al., 1996].

Naval Postgraduate School, Monterey, CA — Some recent embodiment research at NPS focuses on representing real-time arm motion. The egocentric point of view includes a rendering of the user's upper body and arms. In this environment the user, a dismounted infantry soldier, usually carries a rifle. The position of the rifle relative to the user's body is explicitly grounded through the upper body and arm embodiment. The position of a user's arms and rifle relative to the body helps convey intent and activity to other users. For example, an approaching soldier with rifle poised should be attended to in a different manner than a soldier with a rifle by his or her side [Naval Postgraduate School, 1996] [Waldrop et al., 1995].

Swedish Institute of Computer Science (SICS) and Royal Institute of Technology, Stockholm, Sweden — Distributed Interactive Virtual Environments (DIVE) developed at SICS represent users graphically originally as simple blocks and more recently as blocks with limbs and texture mapped faces. Figure 9 shows one incarnation of a DIVE user. The embodiment intends to convey the focus of the user attention, communication capabilities, and activity (e.g., communication capabilities may be inferred through spatial proximity, position, and direction) [Carlsson and Jää-Aro, 1995].

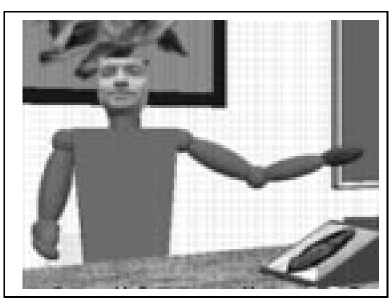

FIGURE 9: User Embodiment in DIVE [http://www.sics.se/dce/dive/]

University of Nottingham, Nottingham, UK — MASSIVE, the prototype VE teleconferencing system developed by Greenhalgh and Benford, employs a spatial model of interaction. Participants are placed in a circle representing a virtual meeting, and users are presented as simple graphical embodiments (see Figure 10 [Bowers et al., 1996]. This representation conveys basic communication cues (position, orientation, etc.) in a simple, efficient manner [Greenhalgh and Benford, 1995].

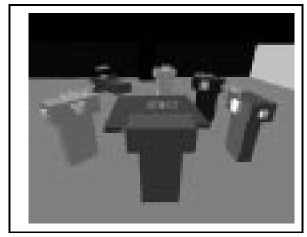

FIGURE 10: A Virtual Meeting in MASSIVE [Greenhalgh, 1997]

6.2.2 VE Agent Representation and Behavior

Another element of virtual models is that of VE agents. Agents have the potential to increase user productivity by performing tasks for the user and guiding the user through learning or complex processes. On the other hand, some agents in VE entertainment have the potential to purposely decrease user productivity, as they are specifically designed to impede the user. The mere existence of agents in VEs may not inherently increase or decrease user productivity. An examination of some VE agent characteristics can help determine whether or not a VE agent is contributing to overall usability in a particular application.

A Virtual Agent

A virtual agent is defined as an entity which "reacts to" and "makes changes in" the virtual world it inhabits. A strict interpretation of this definition fails to address the context and behavior of such "reactions" and "changes." An alternate definition of an agent states that an "agent is a system that can modify its expressive behavior as the context changes and can cooperate with other agents" [Ishizaki, 1996]. In this sense, agents act as performers in an improvisational performance. They rely on their "skills" to react to their environment without any planning strategy in place. By this definition, agents rely solely on context, or system state, with no guiding task goal at hand.

Determining whether or not an agent is ***relevant to user tasks and goals*** ≪Agents1≫ is a possible measure of agent usability. In most settings, agents designed to cooperatively aid the user in a specific task are arguably relevant. On the other hand, there are some circumstances where an agent designed to deter the user from a task may also be relevant. For example, in an entertainment setting, monster agents may be designed to simply search and destroy users. The behavior an agent exhibits within a given context may shed insight into the relevancy of that agent. For example, General Motors (GM) is using VE technology to create virtual car interiors for use on GM showroom floors. Customers are able to experience the "feel" of any GM automobile make and model interior. Agents guide potential customers through a number of selections and choices. In this case, the context is a user-configuration task; the agent reacts by offering help or suggestions. Here, the use of an informed, helpful agent within the context of user configuration is relevant.

Agent Presentation

Agents may appear in a variety of forms: visually, textually, aurally. *Physical realization* is the perceivable representation of an agent [Ishizaki, 1996]. As in user presentation, agent presentation may not be limited to a single form, but instead may be dependent upon context and user resources. In some cases, ***real-world, high-fidelity physical and behavioral representations may be warranted, such as in simulation and training*** ≪Agents2≫. More abstract, low-fidelity forms may be desirable in situations where the user is to be subtly informed, not distracted.

Agent Behavior

Agent behavior is yet another characteristic that can affect usability. Some agents simply react to given circumstances or context, while others may be more goal-oriented. In both cases, the agent's behavior is adapting *dynamically* to situations that arrive. Considering the dynamic and temporal nature of information in most VEs, it may be appropriate to ***allow agents to continuously adapt to changes in context and user activity*** ≪Agents3≫.

Although agents may behave dynamically, their intentions and actions do not have to be too complex to design or understand. Some recent work in agent behavior suggests that developers must be able to ***represent interactions among agents and the "rules of engagement" in a semantically consistent, easily visualizable manner*** ≪Agents4≫ [Trias et al., 1996]. In some cases users may expect agents to follow simple, consistent, "rules of engagement." That is, users may be able to better utilize an agent if they have a clear understanding of how an agent behaves. When virtual agents take on a form users are familiar with, such as a human, a real-world, realistic, interaction may increase suspension of belief and user engagement. Also, by presenting (both physically and behaviorally) virtual human agents as realistically as possible, users are able to tap the wealth of their own real-life experiences, and may be more likely to deduce an agent's purpose, demeanor, etc.

Random agent action and response may confuse users, and deter them from the task at hand. However, there may be times in which non-scripted agent behavior is beneficial, such as in medical training, military training, and entertainment. For example, the knowledge gleaned from a training environment which is too predictable may be prove to be useless in a real-world setting. High Techsplantaions Inc., a medical training and simulation development group, incorporates non-scripted behavior in their surgical simulation environments. Some simulations include one or more "complications" for the surgeon to tend to, while other simulations execute with none. A surgeon who has trained under such circumstances may be better prepared for complications which may arise during a real operation.

Agent Organization

The presence of multiple agents within a VE poses questions about the behavior of the agents as a group. For instance, are the agents working cooperatively or competitively? Are any, or all, of the agents working cooperatively or competitively with the user? If there are multiple agents present, an understanding of the relationship among agents may be useful. For example, are the agents organized in a centralized, hierarchical manner, or a decentralized, lateral manner [Ishizaki, 1996]? Figure 11 gives some examples of these organization styles.

In a centralized, hierarchical organization, the user may convey goals to a single authority agent, who will delegate the lower-level tasks, such as in a military situation. In a decentralized, lateral organization, the user may have to explicitly convey goals to several different agents. An example of this type of organization may exist in an operating room simulation, where the user assumes the role of surgeon. An understanding of agent organization, be it simple (e.g., lateral) or complex (e.g., hierarchical), may aid users in interactions between themselves and agents. If no explicit organization is given, a user will develop their own mental model of agent organization which in turn may or may not be correct. For multi-agent environments, ***agents should be organized according to user tasks and goals*** ≪Agents5≫ [Ishizaki, 1996]. In particular, agents should be organized in a manner such that organization is clear to users and thus allows users to easily delegate tasks and monitor agent progress.

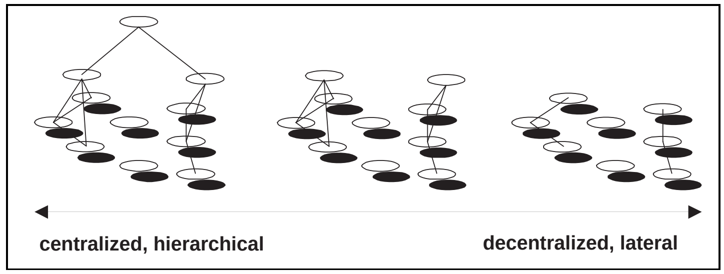

FIGURE 11: Organization Styles [Ishizaki, 1996]

6.2.3 Virtual Surrounding and Setting

The environment in which a VE user is immersed has an effect on usability, by conveying an activity context to the user. That is, a user's behavior within a VE is affected by the environment in which the user is placed. For example, an aircraft cockpit suggests flying a plane and an operating room environment suggests surgical training. In turn, ***a relevant setting may increase user presence and immersion*** ≪Setting1≫ [Barfield et al., 1995]. Furthermore, ***components of the environment may suggest activity as well, via real-world metaphor of functionality*** ≪Setting2≫ [Neale and Carroll, 1997]. Thus, users can expect that a virtual flashlight will provide lighting, a virtual clock will provide the current time, and a virtual phone will provide voice communications. In this case, a real-world, consistent strategy of object appearance and functionality is used to exploit a user's prior knowledge of the real world. This use of metaphor may also decrease the cognitive load associated with translating user intentions to user actions.

Environment Setting

The setting created by the coexistence of objects in a VE may also play an important role in usability. A relevant setting increases user presence by placing the user in an environment specifically designed for a given task. That is, ***a well-designed setting suggests user activity and tasks*** ≪Setting3≫. For example, VE exposure treatment for the fear of heights and flying use settings such as a suspension bridge and the interior of an airborne plane respectively [UNC, 1996] [Hodges et al., 1996]. In some cases, it may be useful to immerse users in an obscure or abstract setting, such as in the SoftImage VE named *Osmose* [Davies, 1996]. One goal of the Osmose VE is to relax users by providing an obscure, non-real world setting in which to explore. "Cartesian notions of space are abandoned for an aesthetic based on transparency, subtlety, and spatial ambiguity" [Davies, 1996]. This creates an environment capable of evoking multiple meaningful associations. The setting appears to increase user engagement and suspension of disbelief evident by user actions and impressions of Osmose. Entertainment VEs are another type in which usability may be influenced by use of a non-real-world setting and physics. "Players" in such VEs may like the ability to do things not possible in the real world, such as fly, morph, teleport, or read thoughts. Furthermore, navigation though large information spaces, sometimes referred to as data mining, may benefit from a non-real-world analogy, in that users may have the ability to search the data space in ways not possible in real-world navigation.

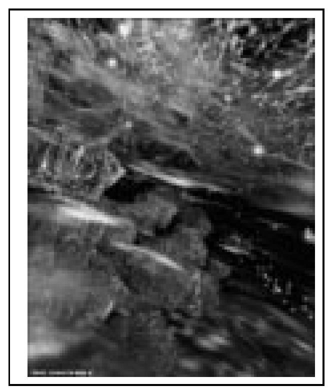

FIGURE 12: An Abstract Setting in Osmose [http://www.softimage.com/]

In some cases, the objects composing a setting should be static, such as instrument and equipment layout in VEs designed to train users on specific equipment. A layout that is constantly changing may be more likely to confuse users both within the VE and when performing the task in the real world.

A dynamic setting may be desirable in VEs designed to mimic continuous events, such as in dismounted infantry simulation. Natural elements such as wind, rain, fog, and temperature may be important to such dynamic environments, yet receive little or no attention in current VE literature.

Fidelity of Objects and Setting

The appearance of rendered objects within the VE is another usability consideration. A high fidelity setting may be desirable in applications which attempt to provide a high degree of realism. Such high fidelity settings can cripple systems when the VE is very large and requires the rendering of a very large number of polygons. Moreover, the high fidelity setting may indirectly trigger usability problems associated with lag and low frame rate simply because computing power is limited. Developers should ***employ rendering techniques that allow for detailed presentation of setting (and other VE elements) without introducing lag*** ≪Setting4≫.

Solutions to the problem of fidelity and computing power include texture mapping, adaptive rendering, and the use of animated video clips. High Techsplanations Inc. (HT), a VE surgical simulator developer, uses texture mapping, image wrapping, flat and gouraud shading to give anatomical objects a realistic appearance [High Techsplanations Inc., 1996]. HT's system also supports variable levels of detail, so that real-time frame rates can be maintained. Head-tracked adaptive rendering may also be used to better balance fidelity and real-time performance [Oshhima et al., 1996]. Objects in an environment are rendered using varying numbers of polygons depending upon their position within a user's field of view. Objects under inspection are rendered in maximum detail. Other objects, such as those on the periphery or those occluded, are rendered with minimal detail. Such an approach takes advantage of situations where users work with a small number of objects within their field of view.

Another method of providing a high-fidelity VE setting uses sequences of video clips animated in response to user input. A video clip is played to give the illusion of forward movement. A user-specified change in direction launches a separate video clip, providing the user a new line of travel as well as a new line of scenery. This method appears to be well-suited for VEs designed to place the user in a real-world location with minimal interaction. For instance, a VE which allows commanders to tour a battlespace may be more concerned with a photorealistic, animated setting, than with interaction between the commanders and objects in the setting.

6.2.4 VE System and Application Information

We use the term "system information" to refer to information which is given "on top of" or "in addition to" information presented to establish environment or setting. Equally as important as environmental information, system information provides users with additional *system state* information such as command interface feedback, navigational aid, and online help.

User Commands in VE Interfaces

A user's view of commands in an interface is one type of information presented as VE system information. Menu look and feel, menu labels, and system messages, to name a few, contribute to an interface's command presentation. Hix and Hartson recommend a simple, yet powerful presentation of command interfaces [Hix and Hartson, 1993]. One method which supports such presentation is the ***use of progressive disclosure: presenting information first at a high level, then revealing more detailed information as users proceed through specific tasks*** ≪SysInfo1≫. Another method used to simplify presentation of commands in interfaces focuses on ***attention to visual organization of the display***. Guidelines ***include eliminating unnecessary information, minimizing overall and local density, grouping related information, and emphasizing information related to user tasks*** ≪SysInfo2≫. Although these guidelines were originally intended for visual 2D GUI interfaces, they certainly may be applied to VEs. For example, a VE employing a rich aural interface may not want to present unnecessary aural cues. Overall and local density of VE aural "objects" should be minimized, so that users do not need to attend to several aural cues at the same time. Related information should be grouped logically and presented as a unit, much like written words.

The look and feel of command presentation, be it visual, aural, or haptic, should be consistent within a single interface ≪SysInfo3≫ [Hix and Hartson, 1993]. This implies that tasks with similar semantics should use similar syntax. One problem with current VE applications is that there is little if any consistency within a single application or among different applications. It is difficult to say whether this is due to the enormous design space that VEs potentially encompass, or whether it is an inevitable result of the lack of an established VE interaction paradigm. Some VE researchers contend that the VE community first needs to establish a set of VE user interface primitives, analogous to 2D GUI objects such as windows, pull-down menus, dialog boxes, etc. [UVA, 1996]. Once defined, these primitives can be used to define a high-level interaction paradigm. Identification and consistent use of such primitives is a basic step for VE user interfaces to reach their usability potential. The other side of the argument would, however, contend that defining such VE primitives could cause a loss of creativity and new ideas in VE user interfaces. Once a set of primitives is established, it may become the de facto standard for VEs, dictating possibly inappropriate design choices to VE user interface designers. If a set of primitives is created, it should be carefully chosen, designed, implemented, and evaluated by users in appropriate application contexts.

What effects would standardized user interface primitives have on VE design space and usability?

Language and labeling for commands in VEs should be carefully designed, clearly and concisely conveying meaning ≪SysInfo4≫. Users typically rely on menu and command labels to infer functionality. Labels presented in a clear, consistent, relevant manner may aid novice users in learning. Obscure or cryptic labels may lead users trudging down frustrating paths or digging through user manuals. For example, an innovative weather visualization environment developed at NCSA uses an obscure domain-specific labeling scheme in a menu, allowing users to select one of many frequency spectrum data sets (e.g., visual, infrared, ultraviolet). Options are labeled as "1", "2", "3", etc. — a short-hand notation recognizable only by meteorologists.

Some command information may be dynamic in nature, providing the user with contextually appropriate information and cues. For example, inappropriate or irrelevant command options should be "grayed out" so that users clearly understand that such commands are not active in the current context.

System Messages

System messages should be ***carefully worded in a clear, concise, constructive manner, so as to encourage user engagement instead of user alienation or frustration or confusion*** ≪SysInfo5≫. Messages should be user-centered, not system-centered. Poorly worded error messages may have negative psychological effects on users, leaving them wondering "what did **I** do wrong." For example, consider the following two messages a VE might present to the user upon receipt of a undefined command:

`''Illegal Command''` and `''Unrecognized Command''`

Displaying the first message may cause users to (consciously or unconsciously) blame themselves, or fear that they were inadequately trying to do something that is against the law. The second message is worded in a less offensive manner, suggesting that the system is to blame. For a more thorough discussion of interface issues in general, refer to [Hix and Hartson, 1993].

Spatial Information

VE system information may also include entities designed to help spatially orient users. [Darken and Sibert, 1995] suggests that the ***presence of a navigational grid and/or a navigational map may have a positive effect on users' ability to perform navigational tasks*** ≪SysInfo6≫. A navigational grid that adheres to the organizational principles (refer to section 5.2.1) may help users understand the organization of a virtual space. The presence of a well-designed navigational map has been shown to help users optimize search strategies, another navigational task examined by Darken and Sibert. One characteristic of a well-designed navigational map is congruency with the environment. That is, users should be able to quickly identify their current position and orientation on the map and in the environment. ***Map design principles*** ≪SysInfo7≫ given in [Darken and Sibert, 1995] infer several other characteristics. These principles include:

1. ***Show organizational elements*** (paths, landmarks, districts);
2. ***Always show the user's position***, so that users can easily relate two points on the map to the corresponding points in the environment; and
3. ***Dynamically orient the map with respect to the terrain and the user*** so that a line between any two map points is parallel to the corresponding line between points in the environment, and so that upward on the map always shows what is in front of the user.

The WIM, previously mentioned, is another example of an entity designed to help spatially orient users. [Stoakley et al., 1995] defines the WIM as "a user interface technique which augments an immersive head tracked display with a hand-held miniature copy of the virtual environment." Figure 13 shows the hand-held miniature which helps users orient themselves by providing a bird's-eye, exocentric view with improved context. Moreover, the WIM may help users with path planning, path history, measuring distances between points, and three-dimensional design. Although the WIM seems well-suited for many spatially-oriented tasks, it seems likely that the metaphor would break down given a very large virtual space. It also seems likely that a high-level, meta-view of a large environment would be too detailed for users to comprehend. A lower-

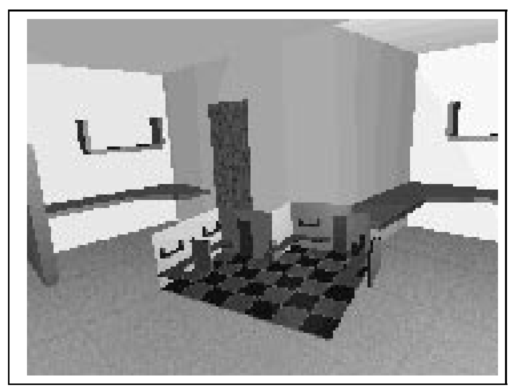

FIGURE 13: Exocentric View of [Stoakley et al., 1995]'s WIM

level view, that shows part of the large VE, would imply some type of scrolling or selection technique, yet still lack the ability to view the entire context.

Application-Specific Information

Another class of VE system information is application-specific, or domain-specific data, i.e., the information the system is designed to deliver. Given the complexity of VEs, the distinction between application data, spatial aids, command interfaces, setting, users, and avatars may not be obvious to users. One usability concern, then, is to ***present domain-specific data in a clear, unobtrusive manner such that the information is tightly coupled to the environment and vice-versa*** ≪SysInfo8≫ [Bowman et al., 1996]. Visualization of complex scientific data (such as DNA or molecular level information) via VEs is only worthwhile if the VE can ***provide additional user insight not available through other presentation means*** ≪SysInfo9≫ (e.g., 2D desktop, mathematically).

7 VE User Interface Input Mechanisms

Engaging in a VE or any other computer-based system implies that some form of dialog exist between user and computer. This dialog, however, is typically not like our natural model of dialog; namely, exchanging spoken words to convey meaning. Instead, the dialog is typically orchestrated through input devices from the user's end, and highlighted, animated displays from the computer's end [Card et al., 1990]. As with any dialog, syntax and content of expression are critical to mutual understanding. An investigation of VE input devices, characteristics, and use may yield clearer comprehension of dialog between users and VEs.

One manner in which to systematically examine input device space is through the use of structured taxonomies [Card et al., 1990] [Foley et al., 1984] [Buxton, 1983]. [Foley et al., 1984] provides useful mappings of input device to tasks, but does not elaborate on device properties that generated the mappings [MacKenzie, 1995]. Moreover, the taxonomies mostly deal with two degrees-of-freedom (DOF) devices such as mice and trackballs, and are marginally extendible to six DOF devices. The above taxonomies are useful for device-task mapping, yet are not capable of predicting and comparing various designs [MacKenzie, 1995]. An exception is [Card et al., 1990], which provides some comparative insight by illustrating how competing designs can be critiqued in terms of expressiveness and effectiveness.

We are not interested in critiquing or creating a taxonomy of input devices per se, but instead interested in characterizing usability issues inherent in input devices and their use. In characterizing these issues, we first look at some general characteristics of VE interface mechanisms: characteristics which can be applied to input devices as a whole. We then delve into a more detailed discussion of specific classes of VE interface mechanisms.

7.1 Characteristics of VE User Interface Input Mechanisms

Although input devices can vary widely in end-use, there are some general characteristics across these devices which may affect usability. Some these characteristics include:

- degrees-of-freedom
- spatial resolution
- sampling rate and lag
- control-display gain
- bandwidth
- resistance (isotonic versus isometric)
- number of users supported
- body-centered interaction ("naturalness" of design and interaction)
- size, weight, comfort, and mobility
- portability
- cost

A general examination of some selected characteristics follows.

Degrees-of-Freedom

By most definitions, VEs and three-dimensional space are one and the same, or at least inseparable. For input devices to provide exhaustive control in three-dimensional space requires that six continuous DOF be addressed. Specifically, this means *translations along* the x, y, and z-axis as well as *rotations around* the x, y, and z-axis. These six DOFs fully define a user's location and orientation. Making the motor-visual leap from 2D desktop devices, such as the mouse, to 3D VE devices, such as the SpaceBall™, is typically hard for most users since humans may not innately *comprehend* three-dimensional space, but instead *experience* three-dimensional space [Hinckley et al., 1994a].

To facilitate ease-of-use in three-dimensional space, VE developers need to carefully examine the manner in which they design control in this space. [Zhai and Milgram, 1993b] presents a framework for studying multi-DOF manipulation schemes which examines the extent to which individual DOFs are *integral* or *separable* [Jacob et al., 1994]. A fully integrated control places all six DOF onto a single device. At the other end of the spectrum, a completely separated control would consist of six individual one DOF devices. The implications to VE system usability are significant, since users must have reasonable control and accuracy in order to perform real work. Thus, VE developers should ***assess the extent to which DOFs are integrabile and separable within the context of representative user tasks*** ≪Input1≫. Assuming that some DOFs must be integrated (manipulating six one DOF devices with at most two hands and two feet would certainly be difficult), the question arises, "which DOFs are inherently integral?" and "which DOFs are inherently separable?"

Before one can assess which of the six DOFs are integral and/or separable, it is worthwhile to consider which DOFs are even required. Most VE applications do not require all six DOFs, and including all six in some fashion or the other may only convolute user interaction. [Hinckley et al., 1994a] suggests that developers ***eliminate extraneous DOFs by implementing only those dimensions which users perceive as being related to given tasks*** ≪Input2≫ [Hinckley et al., 1994a]. For example, in an architectural walkthrough environment (or any other VE where users are restricted to a "walking" metaphor) there may not be a need for all six DOF. Walking about may be modeled as a combination of translations, plus a rotation for pivoting in place, plus a rotation for looking up and down (also useful for plan-views of architectural layout). Thus, this particular task can be performed using five DOFs; inclusion of the additional rotation may simply convolute navigation, rendering the VE harder to use.

However, knowing when to integrate DOFs and when to separate DOFs is not always easy. [Hinckley et al., 1994a] suggests that ***multiple (integral) DOF input is well-suited for coarse positioning tasks, but not for tasks which require precision*** ≪Input3≫. Some telerobotic environments map three DOF to each of two input devices: one for translations and one for rotations [Zhai and Milgram, 1993b]. Other systems may map four critical DOFs (three translations and one rotation) to a hand-held device, reserving the other two DOFs for modal interaction such as during a button press. One approach to determining a good mix is to first understand which DOFs are essential to specific task performance, and then, through user studies, determine which DOFs work well together for the given application.

Once DOF and input device mappings are established, how can developers and evaluators measure the "goodness" of the mapping? Traditional evaluation suggests that metrics such as task time be used to assess interaction. However, for some tasks, time may not be important, such as in surgery. Instead consider an error metric that assesses the amount of *coordination* or control a given mapping affords. [Zhai and Senders, 1997] suggests that spatial deviation in three-space from an "optimal" path is capable of measuring control during positioning. In many cases, control during task performance may not be as important as time to target. However, ***when task performance requires significant coordination and is not time critical*** (e.g., surgery), ***or as simply as a measure of control, consider using "deviation in three-space" as a metric of device control*** ≪Input4≫.

Spatial Resolution

Manufacturers can sometimes give misleading information about device resolution, and typically only give a single, best-case value. As [MacKenzie, 1995] points out, 3D magnetic tracker resolution can drop off significantly when, for instance, the distance between tracker and receiver is increased, two or more trackers are added to the system, and interference from nearby metal objects (e.g., office furniture, shelving) are added. As important as the level of resolution is the trait of *monotonicity*; positive changes in device positioning should always produce positive device output. That is, ***from the user's perspective, device output should be consistent with, and cognitively connected to, user actions*** ≪Input5≫.

Sampling Rate and System Lag

Poor sampling may render an otherwise well-designed VE very difficult to use by introducing lag into the system. Traditional computer device sampling rates are typically 10 to 100 Hz. [MacKenzie, 1995] contends that interactive VEs require much higher sampling rates if they are to realistically model real-world interaction. Moreover, quick user input motions will not be captured accurately if sampling rates are too small.

The traditional tradeoff between low and high control-display gain involves that of gross and fine positioning. ***For fine positioning tasks, employ low gain, for gross positioning tasks, high gain*** ≪Input6≫. Unfortunately, any useful VE will most likely contain both coarse and gross positioning tasks. In these cases, [MacKenzie, 1995] recommends a balance between the two determined by iterative user testing of representative positioning tasks.

Resistance

Although most input devices are thought of as providing uni-directional information *from* users *to* systems, the reality is that information transfer between human muscles and the input device is bi-directional. [Zhai, 1995] points out that this information provides a certain *control feel* and should not be overlooked. Input devices such as tracked gloves provide zero resistance, and are hence termed *isotonic*. Devices which provide infinite resistance such as the Spaceball™ are termed *isometric*. [Card et al., 1991] discusses the relationship between device characteristics (size, shape, etc.) and particular muscle groups (wrist, arm, hand, fingers, etc.). [Zhai, 1995] raises some important questions such as "How do different types of resistance affect user performance?" and "To what extent do different forms of resistance induce user fatigue?" Developers should ***address possible effects that prolonged usage with particular input device(s) may have on user fatigue and task performance*** ≪Input7≫.

What types of user tasks are well-suited for isotonic resistance? For isometric resistance?

Body-centered Interaction

Whether the input device is to be worn, held, or stood upon, designers should strive for natural interaction between user and device. Users of VEs want to engage themselves in worthwhile work or play; user attention should be on the task at hand, and not the device at hand. Thus, in general, developers may be able to ***decrease user cognitive load by avoiding devices such as joysticks and wands which, in effect, place themselves between users and environments*** ≪Input8≫ [Davies, 1996]. A more direct, natural form of interaction may be achieved through glove, voice, and gestural input. In cases where VE model interaction implies some physical implement (e.g., surgery implies scalpel, flight simulator implies flight stick), real-world physical props may be used as opposed to virtual tools coupled with synthetic force-feedback. As mentioned in Section 8.2.3, real-world props serve as excellent input devices when the corresponding real-world tasks explicitly require the use of such tools.

From traditional human factors research we know that ***input devices should make use of user physical constraints and affordances*** ≪Input9≫ [Norman and Draper, 1986] [Hinckley et al., 1994a]. Examples may include placement of buttons on a wand or joystick input device and use of natural, easy to perform gestures. [Hinckley et al., 1994a] also suggests that developers ***avoid integrating traditional input devices such as keyboards and mice in combination with 3D, free-space input devices*** ≪Input10≫ (devices that move freely with users, as opposed to mounted or fixed devices). Users may have a difficult time switching between multiple devices, especially when immersed in an HMD-based environment. Instead, designers should consider adding input capability via voice or gesture input (see Sections 7.2.5 and 7.2.3 for more on these forms of VE input).

7.2 Types of VE User Interface Input Mechanisms

As with traditional 2D computer interfaces, VE input devices are numerous in both function and form. The current momentum (and possibly hype) associated with VE research and development is likely to continue, producing yet more such devices, each with its own unique contribution to user-VE interaction. Although many more devices exist (now and in the future) than can be exhaustively covered herein, we have sampled the major classes of input devices, pointing out characteristics of specific devices which may aid in, or detract from, VE usability.

7.2.1 Tracking User Location and Orientation

One of most fundamental pieces of information a VE system must know is the position of users in three-dimensional space. This position is most often given in terms of location (x,y,z) and orientation (heading, pitch, roll). In many applications, more specific user information is used, such as the location and orientation of users' hands, heads, feet, etc., to create more sophisticated interaction. For example, [Waldrop et al., 1995] describes a method for articulating detailed upper-body movements using magnetic trackers placed on users' wrists, elbows, and shoulders.

Three-Dimensional Position Trackers

Placed on gloves, helmets, body joints, and in hand-held interaction devices, three-dimensional, six DOF trackers are widely used for most every positioning need, and thus may possibly be considered the backbone of VE interaction. Many types of three-dimensional tracking techniques exist, including magnetic [Polhemus Incorporated, 1997] [Ascension Technology Corporation, 1997], mechanical [Sutherland, 1968] [Ware et al., 1993], ultrasonic [Alusi et al., 1997] [Logitech, 1997], and optical tracking [Madritsch and Gervautz, 1996], as well as sophisticated video-imaging techniques [Fukumoto et al., 1992].

A few surveys of tracking technology and design exist [Applewhite, 1991] [Meyer et al., 1992], providing interested readers with good coverage and insight. [Applewhite, 1991] presents an excellent discussion tracker evaluation by means of a framework for suitability. This framework identifies five key measures of tracker suitability including:

1. *resolution and accuracy*,

2. *responsiveness*,

3. *robustness*,

4. *registration*, and

5. *sociability*.

Many of these measures (as well as more detailed measures contained within each item) were discussed above with respect to input devices in general. [Applewhite, 1991] discusses each of these measure, in detail, with respect to specific tracking technologies. Developers and evaluators should ***consider the [Applewhite, 1991] framework when assessing the suitability of tracking technologies with respect to representative user tasks*** ≪Tracking1≫. Complementing the framework are anecdotal evaluations of various tracking schemes which discuss the pros, cons, and suggested use of each scheme. These findings are summarized in Figure 14.

Tracker Lag

As previously discussed, system lag can cripple user interaction by cognitively decoupling user actions and system responses. Trackers are not immune to lag, and thus are potential contributors to overall system lag. Mechanical trackers are capable of reporting position and orientation with significantly less lag than magnetic trackers (as much as 80 msec versus 2 msec [Ware et al., 1993]). This is to be expected, since mechanical trackers rely on mechanical linkage to measure movement, reporting position information through cables. However, the minimal lag associated with mechanical tracking does not come without cost; the mechanic linkage significantly restricts user movements. In general, magnetic tracker lag is not a significant problem and can be managed by minimizing the number of users and trackers used. Indeed, the *vast* majority of VE systems incorporate tracking magnetically with minimal or unnoticeable lag [National Center for Supercomputing Applications, 1996] [University of Washington, 1996] [Naval Research Laboratory, 1997].

Technology	***Pros***	***Cons***	***Suggested Use(s)***
Acoustic	allow multiple sensors to share a single emitter source, some acoustic systems support high data rates within large working volumes	vulnerable to gross errors from sensor occlusion, less significant errors from disturbances in the air	***multi-user systems where high data rates are needed and occlusion can be avoided*** ≪Tracking2≫
Magnetic	good accuracy in small working volumes, not affected by sensor occlusion	poor for open-room applications, vulnerable to electromagnetic noise from power lines and CRTs	***VEs with small working volumes and minimum electromagnetic interference*** ≪Tracking3≫
Mechanical	precise and responsive, largely free from errors	mechanical linkage restricts range of motion, not well-suited for tracking multiple users in a single working volume	***single user applications that require only a limited range of operation, applications where user immobility is not a problem*** ≪Tracking4≫ (e.g., telerobotic applications)
Optical	report high data rates	vulnerable to ranging errors caused by spurious light and occlusion	***real-time applications where occlusion is less likely*** ≪Tracking5≫ (e.g., single user systems)

FIGURE 14: Summarized Evaluation of Tracking Technologies [Applewhite, 1991]

Freedom and Range of User Movement

In order for VEs to become seamlessly integrated into existing work practices, tracking systems must allow sufficient range of motion beyond that provided by current magnetic tracking technology. For VEs that require large user-roaming areas, sophisticated ultrasonic tracking systems may be used to increase user range. Magnetic trackers are typically limited to a range of a few meters, yet do not require line-of-sight. Body-mounted magnetic transmitters are powered through small cables, resulting in some user tethering.

Ultrasonic, optical, and infrared tracking systems avoid tethering and thus allow greater freedom of motion. However, a possible tradeoff is the fact that these systems are susceptible to body interference since line-of-sight is required. Thus, user movements can occlude tracking during task performance, generating spurious data [Strickland et al., 1994].

Very large tracking volumes can be created using a network of trackers and receivers. For example, [Sowizral and Barnes, 1993] describes a tracking system which employs a cellular phone architecture to manage multiple ultrasonic trackers. Although the cellular tracking system is low-cost, while providing high accuracy and reasonably high precision, it is time-consuming to assemble and calibrate. However, for VE applications intended to be used in a fixed (large) location, extensive setup and calibration may be worthwhile.

By definition, user range of motion in mechanical tracking schemes is limited to extent of linkage reach. In some cases this is not a problem, such as in head tracking seated fishtank VE users. Exoskeletons provide mechanical tracking of user limbs, by aligning mechanical linkage between bone joints. While this technique has the potential to accurately track relationships between wrist, elbow, and shoulder, designers of exoskeletons face a challenge similar to that of HMD designers: fitting the various shapes and sizes of users. Unfortunately, most exoskeleton systems are heavy, bulky, and hard to fit to user inter-joint length, resulting in restricted, hindered user motion. For example, demonstrations performed using the SARCOS Sensor Suit revealed that the exoskeleton was cumbersome, difficult to adjust, and moreover produced noisy data [Waldrop et al., 1995].

In general, ***when assessing appropriate tracking technology relative to user tasks, one should consider working volume, desired range of motion, accuracy and precision required, and likelihood of tracker occlusion*** ≪Tracking6≫.

Head Tracking

As previously mentioned, sophisticated VEs require more detailed user position information rather than simply location of the user as a whole. Most VEs employ some form of head tracking for a multitude of reasons: to accurately model egocentric points of view, to render stereographic images with proper perspective, to model motion and parallax cues, to apply gaze-directed selection and rendering. The most common use of head tracking, however, is to support transformation of a VE's visual scene in response to changes in the location and orientation of a user's head [McKenna and Zeltzer, 1995].

[Barfield et al., 1997] writes, "in order to support training and performance in VEs, it is necessary to provide the participant the visual cues (and display hardware) necessary to maintain effective task performance... furthermore, if a sense of presence is beneficial to performance, it also becomes expedient to provide the cues necessary to maintain an appropriate sense of presence." Other research confirms this notion. For example, [Hinckley et al., 1994a] suggests that ***non-immersive, desktop VEs (e.g., VMDs) should employ head tracking techniques as a means to increase user presence by providing head motion parallax depth cues*** ≪Tracking7≫. In general, providing effective visual cues *relative to user motion* relies almost exclusively on head tracking.

Indeed, studies on head tracking generally show that ***head tracking can improve user presence and task accuracy of VMD visualization and manipulation tasks*** ≪Tracking8≫. For example, [Barfield et al., 1997] showed that head tracking and stereopsis were beneficial to task performance and that head tracking had a significant, positive effect of presence. [Snow, 1996] shows that the addition of head tracking increased perceived presence by 61%. [Ware et al., 1993] found that head tracking (coupled with stereopsis) aided in task performance and reduced error rates. Subjective results from the study suggest that head tracking alone is more important than stereopsis alone. [Rekimoto, 1995] found that head tracking (again coupled with stereopsis) increased task accuracy when compared to stereoscopic trials with no head tracking.

How does restricting head tracking to a single user effect collaborative task perfomance?

Other studies have produced similar results using HMD-based VEs. [Pausch et al., 1993] examined the differences between hand tracked and head tracked visual systems. Users performed a search task, locating twenty objects placed in virtual space. Results showed that users controlling point of view via head tracking completed the search task nearly twice as fast, suggesting that ***head tracking is well-suited for search-based tasks*** ≪Tracking9≫. Moreover, performance times were tightly clustered for the head tracking trials, suggesting that the technique can facilitate consistent task performance.

7.2.2 Devices Supporting "Natural" Locomotion

Locomotion by walking may be the most intuitive, natural metaphor available to VE developers. However, implementing the walking metaphor in a natural fashion has proven to be challenging. A number of researchers have developed an array of devices and interaction schemes in attempts to provide this essential interaction, including walking in place [Slater et al., 1995a] [Templeman, 1997a] and the use of modified treadmills [Brooks et al., 1992] [Hirose and Yokoyama, 1992] [Virtual Space Devices, Inc., 1997].

Walking in Place

One might conjecture that the best possible metaphor for natural locomotion would be simply walking about. However, supporting user travel over arbitrary virtual distances while confined to relatively small physical distances is essential for VE success. Thus, while suitable for a very small VE, this metaphor is not appropriate for most. A close match to the natural walking metaphor is that of walking in place. For the purpose of discussion, we do not consider treadmill devices in this section (although users essentially walk in place); instead we consider interaction paradigms that require position tracking hardware only (as opposed to large, moving surfaces). Treadmill navigation is discussed in the following section.

The obvious advantage to the walking in place metaphor is that it requires little or no user cognitive mapping to perform. That is, users simply walk as they would in the real world. Indeed, the walking in place metaphor has been shown to be a very usable means of virtual locomotion. User surveys reported in [Slater et al., 1995a] show that moving through the VE was simpler, getting from place to place was more straightforward, and locomoting through the VE was more natural using the walk in place metaphor as compared to a hand-held 3D mouse. It is interesting to note that users experienced higher levels of presence when using the walking in place metaphor. Thus, ***the walking in place metaphor is well-suited for VEs in which natural locomotion and a high sense of presence are required*** ≪NaturalLoc1≫.

Prolonged virtual experiences using the walking in place metaphor may be tiring for users, especially if each step in the virtual world requires a corresponding step in the physical world. As such, ***the walking in place metaphor may not be well-suited for VEs which require travel over large virtual distances (e.g., exploring extensive models)*** ≪NaturalLoc2≫. Mapping a large virtual step to a small physical step may alleviate the problem to some degree, although the amount of time spent walking in place is more likely to predict fatigue than virtual distance traveled.

Treadmills and Other "Walking" Devices

A number of mechanical devices have been constructed that facilitate "walking" in VEs. Most of these devices are variants on traditional exercise treadmills, where the walking tread moves as users propel themselves forward. That is, the tread is not motorized, but instead moves as users walk. This allows the tread to "track" locomotion speed, including the case where users are standing still. ***For natural locomotion where user roaming is not necessary, treadmill-based locomotion offers a simple, cost-effective solution*** ≪NaturalLoc3≫.

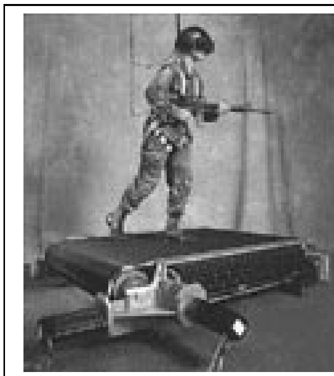

FIGURE 15: VSD Inc.'s Omni-Directional Treadmill [http://www.vsdevices.com/]

Examples of treadmills include the University of North Carolina at Chapel Hill's (UNC-CH) steerable treadmill [Brooks et al., 1992]; which uses a steering bar similar to that of a bicycle; University of Japan's treadmill [Hirose and Yokoyama, 1992]; and Virtual Space Devices, Inc.'s Omni-directional treadmill (ODT) [Virtual Space Devices, Inc., 1997]. The most interesting of the treadmills is the ODT, shown in Figure 15. Two-dimensional roller belts enable users walking or running on the surface to navigate in any direction simply by walking or running in that direction. A harnessing device maintains the user's position at the device center, and simulates the load of climbing and descending hills through pneumatically actuated force feedback.

Another interesting device which claims to facilitate natural walking interaction in VEs was developed as part of Sarcos' Individual Portal (IPORT) [Sarcos Research Corportation, 1997]. The device looks much like a unicycle, since users perch themselves in a seat above a mounted pedaling mechanism. Users control their virtual speed by pedaling, and appropriate force-feedback may be provided to simulate climbing or ascending hills. Direction of travel is determined by seat direction, so that users twist at the waist to perform a virtual turn.

In general, ***supporting natural locomotion through devices such as treadmills and stationary bicycles may impose certain constraints on user movements*** ≪NaturalLoc4≫ [Slater et al., 1995b]. For example, to ensure user safety, users must be suspended, supported, or held to protect against a fall, since immersed users are unaware of potentially hazardous objects that may coexist in the physical setting. These suspension devices are by nature restrictive to certain types of movements (e.g., side-to-side as in turning or leaning, up-and-down as in squatting) and are typically uncomfortable, decreasing user presence and possibly affecting task performance.

Another constraint associated with treadmills is the fact that users must stand and walk in a very small, confined space ≪NaturalLoc5≫ (i.e., on the treadmill surface). Techniques such as walking in place and even hand-based pointing locomotion schemes afford a small amount of lateral movement. This additional (albeit limited) freedom may help increase a user's sense of presence as well as help reduce user fatigue since users are able to move about, shift their weight, and most importantly, not feel compelled to "dismount" a device for relief.

Other safety concerns include the possibility of user injury due to moving mechanical parts. As can be seen in Figure 15, these devices contain a multitude of exposed, mechanical machinery. The majority of these mechanical devices are controlled electronically, posing the potential for electronic glitches, hiccups, or malfunction at the possible expense of users. For example, as mentioned earlier, users of the IPORT change direction by twisting at their waist. The seat is electronically and mechanically controlled to sense the position of a user's waist (to determine current direction). Upon initialization, the unit resets seat position to “forward”, whether or not a user happens to be on the seat! This unexpected, mechanical motion has been known to throw users to the ground. In general, VE developers should ***be wary of any electro-mechanical device of significant size and power since an unexpected malfunction potentially places users at risk*** ≪NaturalLoc6≫.

As with the walking in place metaphor, locomoting in large VEs via treadmill can be tiresome. Moreover, users may become frustrated at the slow pace imposed by this natural walking technique [Brooks et al., 1992]. Thus, ***treadmills may not be well-suited for VEs which require travel over large virtual distances*** ≪NaturalLoc7≫.

7.2.3 Data Gloves and Gesture Recognition

Someone who has been asked to describe a VE will typically include two major devices in their response: an HMD and a data glove. No other input device is so closely connected with the perception of VEs. A natural extension of human behavior, gloves not only allow VE users to reach, grab, and touch virtual objects of interest, but to engage in gestural interaction (e.g., pointing to an object as a means of selection).

Gloves

Natural VE interaction directly through the hands (as opposed to through devices) should be, and is generally, a goal of many VE researchers. [Sturman and Zeltzer, 1994] writes, “clumsy intermediary devices (3D mice, joysticks, etc.) constrain our interaction with computers and their applications... glove-based input devices let us apply our manual dexterity to the task”. One advantage that data gloves certainly have over 3D joysticks, SpaceBalls, etc., is the ability to recognize hand gestures; a powerful distinction as we strive to develop deviceless VE user interfaces.

As specialized input devices, gloves typically report the position hands in 3-space, and more importantly, the position of the fingers relative to the hand or palm. Most gloves report six DOF location and position information through magnetic trackers mounted on the back of the glove. Thus, the ***DOF issues discussed in Section 7.1 are applicable***

to glove-based input ≪Gloves1≫.

To measure finger position relative to the hand, most gloves are equipped to capture finger-joint position through flex sensors. There are generally two schools of thought on capturing these positions: (1) through optical or electronic channels mounted *within* the glove, and (2) through mechanical linkage mounted *outside* the glove (a.k.a. exoskeleton). In either case, to capture the most basic hand and finger positions, gloves typically use two flex sensors per finger (used on the lower two knuckles). More sophisticated designs capture flexion in the distal joint (finger's outer most knuckle) for more detailed gesturing.

What degree of glove precision and complexity is needed for a given set of user tasks?

The CyberGlove by Virtual Technologies [Virtual Technologies, Inc., 1997] is considered to be one of the best gloves on the market, possibly because of its additional sensing capabilities beyond those described above. In addition to measuring finger joint flexion, the 22-sensor model also measures the angle between adjacent fingers (abduction), the extent of thumb crossover, palm arch, wrist flexion, and wrist abduction ***Understanding the complexity of representative glove-based tasks (with respect to finger, thumb, and wrist flexion) may aid designers in identifying appropriate glove designs*** ≪Gloves2≫.

Fakespace's Pinch Glove™ is capable of reliably recognizing basic gestures without the additional cost incurred by sophisticated flex sensors [Fakespace, 1997]. Each glove contains five electronic sensors (one in each fingertip), designed to be used in pinching combinations. Contact between any two or more digits completes a unique electrical path that is then mapped to an application-specific meaning. Multigen™ has successfully developed an entire language of gestural "pinching" for use in its SmartScene packages [MultiGen, 1997]. ***Very natural gestural interaction may be achieved through intuitive pinch mappings*** ≪Gloves3≫. For example, "pinching with forefinger and thumb" may used to grab a virtual object and "snapping between middle finger and thumb" may used to initiate an action.

Spatial Resolution of Flexion

Much like other mechanical tracking systems, exoskeleton-like gloves afford the highest resolution at the expense of comfort. For example, the Exos Dexterous HandMaster is capable of reporting finger flexion within one degree, making it, and other ***exoskeleton-based gloves, well-suited for very fine-grained manipulation tasks*** ≪Gloves4≫ [Exos, 1997]. This advantage has also granted the HandMaster (and others like it) an interesting niche in the medical community, accurately measuring hand and finger movement of those recovering from hand or nervous-system injury.

Data gloves which have limited finger flex accuracy (5-10 degrees) may not be well-suited for complex recognition or fine manipulations ≪Gloves5≫ [Sturman and Zeltzer, 1994]. The CyberGlove, on the other hand, claims to support finger flex accurate to 0.5 degree, remaining constant over the entire range of joint motion, making it a likely candidate for such tasks [Virtual Technologies, Inc., 1997]. Gloves with low sampling rates, typically around 30 Hz, may not be able to capture very rapid hand and finger movements such as those used in time-critical applications [Sturman and Zeltzer, 1994]

Other Usability and Human Factors Issues

Exoskeleton-based gloves are typically cumbersome to take on and off and require adjustment to fit a user's hand properly. Moreover, the exoskeleton-based gloves can weigh up to 12 ounces (four times as heavy as optics-based gloves), making the glove unstable, especially when the hand is shaken or moved rapidly [Sturman and Zeltzer, 1994]. Thus, ***exoskeleton gloves are not well-suited for general or casual use*** ≪Gloves6≫.

[Virtual Technologies, Inc., 1997] claims that "the CyberGlove provides high quality measurements for a wide range of hand sizes, and ensures repeatability between uses." Although calibrations do not need be updated after every use, tests with repeated use show that variation in resolution is approximately one degree of standard deviation. The Pinch Glove™, on the other hand, does not require any calibration and works with most sized hands.

Gesture Recognition

Sophisticated gestural interaction has the potential to transform today's clumsy, device-driven VE interfaces into ones of natural and intuitive expressiveness. Indeed, most of the gestures needed for an intuitive interface exist in our current western culture; pointing to select, curling the index finger towards oneself to call someone closer, crossing one's arms to show discontent or disapproval, etc. The obstacles to achieving this goal are mostly grounded in reliable, flexible, gesture recognition.

Recognizing gestures starts by understanding and interpreting the relationships among body posture, intended meaning, and intended result. Although recognizing full body gestures (arm position, stance, etc.) and facial gestures has the potential to facilitate a complete, gestural interface, the majority of current research focuses on hand gestures (a smaller, yet useful subset of gestures).

For example, [Sturman, 1992] classifies three types of interpretations of hand actions for use as computer-based input: direct, mapped, and symbolic. Direct interpretation, the most basic interpretation, is essentially "point, reach, and grab" interaction. Mapped interpretation refers to "fiddling" with devices at a finer-grained level. Symbolic interpretation, the most abstract level, recognizes hand gestures as a stream of semantic tokens. To date, most VE hand-gesture interaction can be classified as either direct or mapped, using the index finger to point, etc. However, as [Su and Furuta, 1994] points out, these interactions do not exploit the full range of expression available through hand gestures, and in fact, can typically be reduced to basic functionality available through 3D mouse devices and the like. Thus, ***to fully realize the power of natural gestural interaction, VE systems need to recognize a particular sequence of hand postures (tokens) as something more than simply the sum of the parts*** ≪Gesture1≫.

Unfortunately, gesture recognition is difficult since most gestures are continuous expressions with one leading into the next and so on. As such, the set of available gestures is typically predefined. That is, users must *learn* a prescribed set of gestural commands. Although suitable for pioneering research, there is a need to move past this, and begin developing systems that ***allow gestures to be defined by users incrementally, with the option to change or edit gestures on the fly*** ≪Gesture2≫ [Su and Furuta, 1994].

There are two main approaches, from a system perspective, to recognizing gestures in real-time: glove or body-based recognition [Wexelblat, 1995] [Jacoby et al., 1994] and image processing-based [Maggioni, 1993] [Bröckl-Fox et al., 1994]. Although both paradigms are capable of reliably recognizing hand and body gestures, there are some differences with respect to real-world use and usability. One difference is the paradigm's effect on working volume. Image processing-based gesture recognition imposes a limited working volume on users, typically the volume of a desktop monitor. Thus, users are "gesturally tethered" to a fixed working volume. For this reason ***image processing-based gesture recognition is well-suited for desktop or fishtank VEs*** ≪Gesture3≫. Moreover, since ***image processing-based recognition*** requires dedicated cameras and line of sight, it ***is more appropriate for single user VEs*** ≪Gesture4≫.

Glove-based recognition, on the other hand, allows for a much larger user working and roaming volume. Although tethered by wire, users' gestures are recognizable anywhere within the tracking space, and assuming magnetic tracking, without the risk of occlusion. This makes ***glove-based recognition systems well-suited for VEs which allow some degree of roaming*** ≪Gesture5≫, such as CAVEs™ and HMD-based systems. Moreover, since glove-based gesture recognition is by definition glove-centered, and hence, user-centered, supporting multiple user interaction gesturally is possible.

For user-centered gestures to be truly "user-centered", recognition systems must define a frame of reference relative to each user. Performing gestures with no frame of reference is difficult since gesturing attributes such as distance and direction have no basis. As such, developers should ***avoid gesture in abstract 3D spaces, and instead use relative gesturing*** ≪Gesture6≫ [Hinckley et al., 1994a]. For example, two-handed interaction is generally considered a usable interaction technique in part because it allows users to make gestures relative to each hand. To use a palette menu (a two-handed interaction technique), users hold the menu in one hand (the non-dominant hand) and select menu options with (e.g., the finger of) the other hand (dominant hand). The distance from the start of the gesture to the palette surface is easily perceived by users, since they are essentially bringing the finger of one hand into the palm of another. For more detailed information on the advantages of two-handed interaction see Section 5.2.3.

7.2.4 Magic Wands, Flying Mice, SpaceBalls, and Real-World Props

Like many other devices used in VEs, the development of hand-held input devices has mainly been gadget-driven as opposed to human-driven. That is, it has been driven by technological advances (in the case of input devices, sensing technology) as opposed to existing knowledge of human characteristics and performance [Zhai, 1995]. As with data gloves, these devices typically provide six DOF location and position information through mounted magnetic trackers or through physical manipulation. Thus, the ***DOF issues discussed above, such as eliminating extraneous DOFs, and the integrability and separability of DOFs, are directly relevant*** ≪HandHeld1≫.

Designers wishing to integrate six DOF control have a multitude of devices to choose from. A theoretical approach to understanding input device space is available through the use of a framework of six DOF input [Zhai and Milgram, 1993b](later revised in [Zhai and Milgram, 1994]). While this framework was initially developed to structure investigative research, it can certainly be used as a framework for discussion and comparison of existing input devices. The three-dimensional framework, shown in Figure 16, assigns sensing mode (mapped from *isotonic* to *elastic* to *iso-*

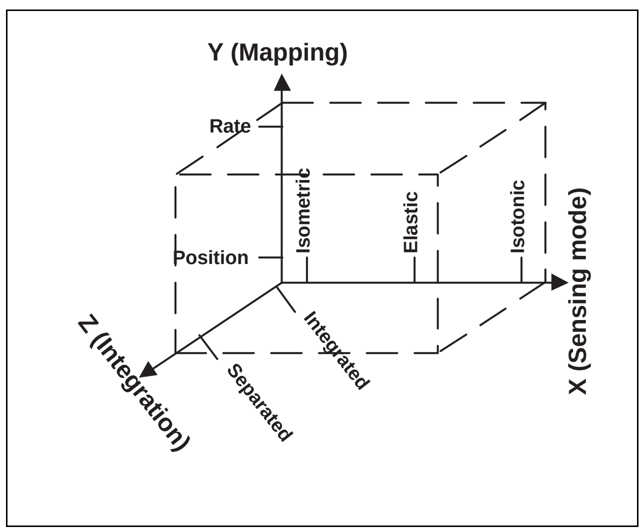

FIGURE 16: A Framework for Studying Six DOF Input [Zhai and Milgram, 1993b]

metric), mapping relationship (*position* vs. *rate*) control, and degree of integration to x,y, and z axes respectively.

As previously mentioned, isotonic devices are thought of as providing zero resistance, such as a data glove, and isometric devices are thought of as providing infinite resistance, such as a SpaceBall™. [Zhai and Milgram, 1994] has also examined the use of *elastic* devices, such as the Elastic General-purpose Grip (EGG). The [Zhai and Milgram, 1993b] (later revised in [Zhai and Milgram, 1994]) framework situates elastic devices between isotonic and isometric devices, although it is closer in most respects to isometric controllers [Zhai and Milgram, 1993a].

Another important characteristic of input devices explicitly addressed in the [Zhai and Milgram, 1994] framework is that of mapping relationship; that is, the mapping between user limb and resulting cursor or object movement. [Zhai and Milgram, 1993b] describes two common mappings: *position* and *rate* controlled. Position mapping essentially maps user limb to cursor or object position by a pure gain. That is, a change in a user limb's x,y,z position and orientation results in an absolute, proportional change in cursor or object x,y,z position and orientation. Rate controlled schemes, on the other hand, use time integration to map user limb position to cursor or object position. For instance, a user gesturing forward would cause virtual movement in a forward direction, proportional to the extent of user limb movement.

Desk-mounted controllers such as the SpaceBall™, Logitech Magellan SpaceMouse™, and the SpacePuck™ are isometric devices which can be programmed to map user movements into either position- or rate-controlled schemes. Free moving devices that users hold or wear, such as gloves mounted with trackers, real-world props, CAVE™ and ImmersaDesk™ Wands, the Cricket™, and Colin Ware's Bat, are isotonic devices, which can also be implemented as either position- or rate-controlled devices.

Perhaps the most interesting results of this work are found in the numerous comparisons of devices within the framework space. For instance, [Zhai and Milgram, 1993b] compares four input device implementations (isotonic position, isotonic rate, isometric position, and isometric rate) within the context of six DOF object manipulation and positioning. The isotonic position and rate trials used a simple data glove equipped with an Ascension Bird™ tracker, while the isometric trials used a SpaceBall™ . Initial results indicated that isotonic position control and isometric rate control allowed for faster trial completion times than the other two methods, suggesting that ***free moving, isotonic input devices (such as tracked gloves) may be more useful when implemented as position controllers*** ≪HandHeld2≫, and that ***desktop, isometric input devices (such as the SpaceBall™) may be more useful when implemented as rate controllers*** ≪HandHeld3≫.

Follow-up research compared an elastic rate controller (EGG) to an isometric rate controller (SpaceBall™) [Zhai and Milgram, 1994] [Zhai and Milgram, 1993a]. These studies show that elastic rate controllers result in faster performance time and lower tracking errors in both simple and more complicated target positioning and tracking tasks, suggesting that ***elastic rate controllers are well-suited for object manipulation and positioning tasks*** ≪HandHeld4≫.

In another comparison of input devices, [Zhai et al., 1996] examines two isotonic, position controlled devices to understand the effect of certain muscle use on task performance. Results from the study show that performance of six DOF docking tasks may be performed faster and more accurately with a fingerball (a small hand-held device manipulated by fingers only) as compared to a glove with mounted tracker. Thus, it may be the case that ***small, hand-held devices that exploit the bandwidth of human fingers may have performance advantages of devices relying on larger muscle groups, such as gloves*** ≪HandHeld5≫. Other implications of the work suggest that ***assessment of six DOF input devices include the degree to which size, shape, and use of device affords manipulation with fingers as opposed to larger muscle groups (e.g., wrist, forearm, shoulder)*** ≪HandHeld6≫.

An advantage of ***desktop, isometric devices*** is that they ***are typically not worn, thus facilitating ease of device integration into working, desktop environments***

(e.g., CAD environments where users switch between six DOF device and keyboard, etc.) ≪HandHeld7≫. Another advantage that some isometric devices afford is multimodal interaction. That is, ***isometric input devices may be coupled with haptic feedback to facilitate bi-directional information flow between user and object, resulting in a more natural interaction*** ≪HandHeld8≫. For example, there are many haptic joysticks commercially available that serve as both input and output devices.

Real-world Props and Tools with Mass

Real-world props (a.k.a. tools with mass) ***have been shown to be an intuitive and powerful form of VE input (and output)*** ≪HandHeld9≫ [Badler et al., 1986] [Hinckley et al., 1994b] [Stoakley et al., 1995]. Under the interaction technique paradigm [Hinckley et al., 1994a] refers to as "physical manipulation", real-world objects, typically used or related to domain-specific objects, are instrumented with trackers so that users may physically manipulate an object corresponding to some visually rendered virtual object. For example, the Naval Research Lab (NRL) is using an actual fire hose nozzle and control end as an input device to a VE designed to train fire fighters on ships.

Both [Hinckley et al., 1994a] and [Hinckley et al., 1994b] enumerate advantages of the physical manipulation interaction paradigm, and in particular, real-world props. Perhaps one of the major advantages is the fact that ***real-world props allow "the computer to interact with the real environment controlled by the operator"*** ≪HandHeld10≫ [Badler et al., 1986]. As [Hinckley et al., 1994b] points out, this forces the *computer* to translate the user's input stream, as opposed to the user translating their physical manipulations.

Similarly, presence of ***real-world props or tools with mass enable bi-directional information flow inherent in complex user-VE interaction*** ≪HandHeld11≫. As output devices, they provide natural, haptic feedback through weight, shape, hardness, etc. (see Section 8.2.3 for more on real-world props as output devices). Advantages relative to the use of real-world props as input devices include:

- mass of the prop or tool "damps instabilities" in the user's hand motions;
- physical properties of the prop or tool suggest its use and how it can be manipulated;
- manipulating real-world objects is a familiar task and exploits existing user knowledge and skill;
- users are immediately and continuously aware of the physical existence of the prop (as opposed to merely a visual representation which may be cluttered among other visual elements); and
- developers can provide familiar tools to user tasking, rather than an interface limited by the physical interaction of gloves, joysticks, or desktop six DOF devices.

For instance, informal observations by [Hinckley et al., 1994a] note that users of a search-task VE found glove-based manipulation of a virtual flashlight inordinately difficult when compared to the ease-of-use afforded by a real flashlight.

Related research from the University of Virginia describes the success of integrating real-world props into the World in Miniature's (WIM) user interface [Stoakley et al., 1995]. Prior to use of a real-world prop as a physical representation of the WIM, users would "contort themselves into uncomfortable position" when attempting to orient the virtual WIM. However, when a magnetically tracked physical clipboard was included in the interface, users had no problem rotating the WIM, interacting naturally with two hands to perform the task. Despite the apparent usefulness of real-world props, there are potential problems associated with their use. For example fatigue may be a problem in VEs that include moderately weighted props. ***Fatigue associated with use of real-world props may be reduced by including some type of clutching mechanism*** (see Section 5.2.3) ***or by providing proper arm support*** ≪HandHeld12≫ [Stoakley et al., 1995]. Another potential problem identified by [Stoakley et al., 1995] is that moderately sized props may occlude virtual objects or the entire virtual display as users are prone to holding the props close to their eyes.

Some Examples of Six DOF Input Devices

We present three examples of six DOF input devices simply to illustrate the continuum of sensing modes described in [Zhai and Milgram, 1993b]'s framework (i.e., isotonic, elastic, and isometric).

The Wand — Isotonic Device

Both the CAVE™ and the ImmersaDesk™ use the University of Illinois at Chicago Electronic Visualization Laboratory's (EVL) wand as their main interaction device (see Figure 17). Essentially a 3D mouse, the wand is a television remote-sized device containing a six DOF tracker. To complement position and orientation information, the Wand contains three buttons and a pressure-sensitive joystick that can be programmed to serve a number of uses. The joystick is used primarily for navigation while the buttons are primarily used to set modes and select options [University of Illinois at Chicago Electronic Visualization Laboratory, 1997]. Direction of travel is typically specified by wand orientation, as opposed to user gaze [Sowizal, 1994]. This allows users to travel in one direction while looking in another (much like we are able to do in the real world).

FIGURE 17: EVL's Wand

[http://evlweb.eecs.uic.edu/EVL/VR/]

Zhai's Elastic General-purpose Grip (EGG) — Elastic Device

The EGG is simply a tracked, symmetrical, egg shaped object that is elastically mounted in three-dimensions (see Figure 18). The elasticity of the EGG facilitates self-centred, rate control interaction (as opposed to position control). Used in many of Zhai's experiments as an instance of an elastic device, the EGG has been shown to provide smooth, noise-free use with minimal effect on user fatigue. For example, subjective ratings from [Zhai and Milgram, 1993a] show that users prefer elastic rate control (via the EGG) over isometric rate control (via the SpaceBall™). Moreover, users found the EGG easier to use with less associated fatigue that the SpaceBall™.

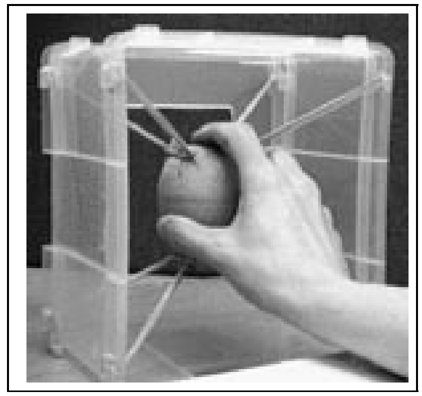

FIGURE 18: Shuman Zhai's Elastic General-purpose Grip (EGG) [Zhai, 1997]

The Spaceball™ — Isometric Device

The SpaceBall™ has proven itself a continued leader in the desktop isometric market. SpaceBall™ makers claim the device to be fast, easy and intuitive to use, providing smooth, dynamic simultaneous six DOF control. Virtual cursor or objects are manipulated simply by lightly pushing, pulling, or twisting the SpaceBall™.

FIGURE 19: Spacetec IMC's Spaceball 3003 [http://www.spacetec.com]

The SpaceBall™ is designed to allow a user's hand to rest in a very "natural, relaxed position on the ergonomically designed base, eliminating arm or hand stress and fatigue" [Spacetec IMC, 1997]. Moreover, makers of the SpaceBall™ claim that the optimal shape of the hand-held molded ball facilitates precise, intuitive control because its symmetrical shape offers no "preferred" axial direction.

7.2.5 Speech Recognition and Natural Language Input

Robust and reliable speech recognition has been an ongoing challenge for computer researchers and engineers. Likewise, integration of natural language into computer interfaces has been an ongoing problem since early investigation with keyboard-based dialog. Although the two can be, and have been, examined separately, the advancement of voice-based input for VEs may rely on a synergistic union of natural language input *through* speech recognition. Unfortunately, there are too many specific design issues to properly cover in this taxonomy, and as such, we attempt to keep the discussion brief, focusing on issues which potentially effect user performance.

One the most promising uses of speech recognition is that of complementing other VE input modalities; creating powerful multimodal interfaces. For example, ***combining speech input and pointing may result in more usable selection mechanism*** ≪Speech1≫ (as compared to kinesthetic or physical manipulation), especially in VEs where objects are distant or tightly clustered [MacKenzie, 1995]. Combining speech and gesture has been shown to be a valuable interaction strategy as well. For example, the combination of speech and gesture has been shown to be useful for object manipulation tasks, such as rotation, translation, and scaling of objects. Users in the [Hauptmann, 1989] study, when given the ability to use voice, gesture, or combination input, unanimously chose to integrate the two modalities. [to the Interface, 1996] enumerates two other advantages of combining the two modes including the fact that gesture and voice naturally complement each other since natural language is well-suited for descriptive input, and gesture is well-suited for direct manipulation and the fact that combining speech and gestural input improves system recognition. Speech recognition systems alone are typically unable to handle the large variety of input needed to give VEs full functionality [Sowizal, 1994]. Thus, ***VE developers should use speech recognition and natural language input as a complement to multimodal interfaces, as opposed to stand-alone mechanisms*** ≪Speech2≫.

Natural speech recognition can make VEs easier to use be offering the ability to make more direct charges to the environment ≪Speech3≫. For example, consider the difference between altering a virtual object's color and shape via virtual tools versus via verbal commands in a VE designed to support interior design. While users could use virtual tools such as a virtual saw and virtual paint to alter a table's appearance, it would be much easier and faster to issue the verbal command, "paint the table red and make it round" [Karlgren et al., 1995].

[Karlgren et al., 1995] makes an interesting observation regarding the nature of human-computer speech interaction. In particular, it is observed that the prominent metaphor used in VEs places users in virtual worlds with no obvious entity to verbally address. The implication for speech recognition is, that users are left to converse with the computer; a disconcerting notion for most people that may detract from virtual presence. Instead, ***VEs supporting speech interaction should include a proxy, agent, or god-like entity for users to verbally address*** ≪Speech4≫ [Karlgren et al., 1995]. For instance, SRI International has developed a "Simon says" metaphor to facilitate user-computer dialog, where users address a god-like entity with the name, "Simon", followed by a verbal command.

<u>Reference Resolution</u>

Some of the difficulties with natural language recognition are centered around the lexical, syntactic, and semantic ambiguities inherent in languages [Hix and Hartson, 1993]. For example, the statement, "bring that here", is semantically vague since the word "that" may refer to one of many objects in the scene. Reference resolution, as the term suggestion, is the process of resolving ambiguous language references such as "that", "this", etc. Optimally, a language recognition system's reference resolution algorithm will consistently identify the object(s) or concept(s) of discourse.

Unfortunately, even the most sophisticated reference resolution algorithms will not always yield the correct result. [Cohen, 1992] suggests that this may be due in part to underdeveloped knowledge bases and the fact that the system typically does not have access to the specific discourse situation. However, the development of a system which can accurately and consistently resolve ambiguous references in one attempt is unlikely, and as such, ***VEs should strive to support* incremental *human-computer discourse*** ≪Speech5≫. [Karlgren et al., 1995] points out that “users of computer systems do not expect the system to manage discourse, but instead, take full responsibility for the coherence of a discourse.” Thus, VEs need to support incremental discourse by providing feedback to users such as what information is available for reference, and what assumptions the system is making.

[Karlgren et al., 1995] further argues that VEs need to provide feedback regardless of reference resolution outcome. That is, an incorrect outcome is not necessarily bad, as long as users are made explicitly aware of the outcome. Without this feedback, users may become confused, wondering perhaps if their verbal command was not understood, not executed, or executed outside their field-of-view.

Limitations and Drawbacks of Speech Recognition

One of the well known limitations of command-based speech recognition systems is that a significant amount of cognitive overhead is required by users to track and recall commands. Menu-driven interfaces offer visual cues to aid user recall, yet require explicit physical manipulation to navigate. An optimal speech-based interface would allow *users* to dictate and develop discourse semantics. As such, ***a sophisticated natural language recognition system should “learn” user syntax and semantics so that the computer interprets user language*** ≪Speech6≫ as opposed to the other way around.

Another limitation of today’s speech recognition systems is that they typically have to be “trained” to each individual user, sometimes taking up to 20 minutes to complete. Even when trained, these systems may not be able to distinguish one speaker from another, making multi-user scenarios quite challenging. In such cases, the use of head-mounted microphones may be helpful in isolating individual speakers.

Annotation

One of the more promising benefits of integrating speech recognition into VEs is that of speech annotation. That is, tagging virtual scenarios, objects, events, or even other users with verbal commentary. As [Harmon et al., 1996] points out, ***verbal annotation is useful for applications areas, such visualization, simulation, and training VEs, where preserving contextual information is important*** ≪Speech7≫. For example, users of a VE designed for scientific visualization may wish to capture analysis remarks related to specific data sets. Likewise, verbal annotation to simulations provides additional context when the simulation is played back (e.g., for the purposes of training or evaluation).

One of the first systems to provide voice annotation in VEs allowed users to annotate objects by first selecting objects via tracked mouse, then pressing a mouse button to initiate the annotation, and then recording the annotation [Verlinden et al., 1993]. A visual annotation marker then appeared on the object, allowing users to playback annotations at will. The Virtual Annotation System, or VAnno, developed at the Georgia Institute of Technology, extends this model into a more robust set of annotation and playback capabilities. [Harmon et al., 1996] enumerates some of the important features of VAnno, which may be interpreted as potential requirements of integrating voice annotation into VEs, including:

- ***strive for seamless integration of annotation into VEs, requiring no mode switching to record*** ≪Speech8≫,
- ***provide quick, efficient, and unobtrusive means to record and playback annotations*** ≪Speech9≫, and
- ***allow users to edit, remove, and extract or save annotations*** ≪Speech10≫.

8 VE User Interface Presentation Components

VEs rely on specialized hardware to "present" information to users. Note that we use the term "present" and "presentation" to imply much more than simply a visual context — all the senses can be used in VE user interface presentation. Section 6 discussed the types of information that may exist in a virtual model, and thus, are subject to presentation. We now discuss the devices, or *presentation components*, used to support presentation. A system's presentation components may have an effect on a user's cognitive processes (among many others), and subsequently, usability.

8.1 Characteristics of VE User Interface Presentation Components

Historically, hardware associated with VE presentation has been specialized to render a single facet of human senses; HMDs were developed for visual rendering, 3D localized sound systems for aural rendering, force feedback devices for haptic rendering, etc. Given the complexity of human sense and perception, it is no surprise that device researchers and developers focus on single sense presentation (with some exceptions such as simple 2D sound integrated into headsets). Although these interface components enable the rendering of different, separate sensory information, they share common characteristics such as the following:

- dimension rendering
- spatial resolution
- refresh and update rates
- intensity
- range
- bandwidth

- number of users supported
- "naturalness" of design and interaction (body-centered interaction)
- size, weight, comfort, and mobility
- portability
- cost

A discussion of these characteristics as they apply to individual component classes (e.g., visual, aural, haptic) follows.

8.2 Types of VE User Interface Presentation Components

The following sections discuss classes of presentation components with an emphasis on usability characteristics rather than specifics of brand name products. This approach is generally more applicable, as an understanding of component types, characteristics, and uses helps VE designers evaluate, acquire, and/or integrate any piece of VE hardware, new or old, regardless of brand name.

8.2.1 Visual Feedback — Graphical Presentation

The roots of VE displays can easily be traced back to development of the television. However, it was pioneers such as Ivan Sutherland who took the first big steps by developing three-dimensional HMDs [Sutherland, 1968]. Nearly thirty years later, VE researchers are still refining and inventing various types of displays used to present visual information to users. The field has grown to include not only head-mounted displays, but spatially immersive CAVEs™ and desktops, virtual modeling workbenches and desks, and even virtual retinal displays. Commercial developers aggressively pursue next-generation displays which are lighter, exhibit greater field of view (fov), and provide better resolution. Yet we are just beginning to understand emerging relationships between display device characteristics and user tasks. Mapping user scenarios and tasks to appropriate display types and characteristics is essential for the development of truly useful and usable visual displays.

Although the visual presentation arena appears to be diversifying, some common display characteristics persist. A review of VE display literature reveals a large number of such characteristics, most of which are technical specifications. Of particular interest are those specifications which have repeatedly been shown to have measurable effects on user task performance, user error rates, and learning. These include stereoscopic support, spatial resolution, fov, update rates, and refresh rates. Non-technical characteristics such as the number of users supported, user comfort, and user acceptance are equally important yet receive much less attention.

Stereopsis

VE users who wish to experience a truly visual three-dimensional world rely solely on a display's ability to generate stereoscopic images. In VE, *stereopsis* refers to stereoscopic vision, exploited by generating and presenting slightly different images to each user eye; the net effect is a *perceived* three-dimensional scene. With all the hype surrounding VEs one would believe that stereopsis is an essential, required characteristic. This is not the case because some VE tasks are better suited for monoscopic displays. Some current research on stereopsis aims to distinguish stereoscopic tasks from monoscopic tasks. For example, based on literature review, [Davis and Hodges, 1995] enumerate some conditions and situations where ***stereoscopic displays are beneficial*** in enhancing perception and task performance. Such conditions include the following:

- ***when information is presented in an*** **egocentric view** ***rather than an*** **exocentric** ***view*** ≪Visual1≫;
- ***when monocular cues are*** **ambiguous** ***or*** **less effective** ***than stereoscopic cues*** ≪Visual2≫;
- ***when presenting relatively*** **static** ***or*** **slowly changing** ***scenes*** ≪Visual3≫;
- ***when presenting*** **complex** ***scenes,*** **unfamiliar*****, or*** **ambiguous** ***objects*** ≪Visual4≫;
- ***when 3D manipulation tasks require*** **ballistic** ***movements*** ≪Visual5≫; and

- ***when user tasks are* highly spatial** ≪Visual6≫ (e.g., precise placement of tools, 3D docking, visual searching)

When appropriate, stereoscopic displays may be beneficial to system usability. For example, stereoscopic displays afford finer discrimination of depth and distance between objects than monoscopic displays. Furthermore, users tend to perceive spatial aspects of the environment faster and more accurately [Davis and Hodges, 1995]. It is likely these characteristics ultimately lead to reduced task errors, reduced learning time, and increased task performance, all of which stereopsis has shown to affect in a positive manner [Drascic, 1991]. For example, a 1997 study on stereoscopic displays showed that stereopsis increased perception of shapes and spatial orientation [Barfield et al., 1997]. Users were given a virtual bent wire to inspect in three dimensions, and then had to select the corresponding two-dimensional representation of the bent wire. Average response accuracy for the task increased from 40% to 65% when stereopsis was added to the representation.

Head tracking and Field of View

When visual displays involve a head-mounted device, such as in HMDs and spatially immersive displays (SIDs), fov and display should be coupled via head tracking ≪Visual7≫. The coupling of fov, display, and head tracking has been shown to reduce task completion time as well as learning time when compared to a hand-tracked environment [Pausch et al., 1993]. Head tracking stereoscopic SID images is necessary due to the nature of presentation. In this case, shutter glasses (worn on the user's face) are tracked so that the left and right images can be rendered appropriately. In domes or VE theaters, users are generally stationary so that head tracking is not always necessary. A more detailed discussion of head tracking as a critical input mechanism is given in Section 7.2.1.

Update, Refresh, and Lag Rates

Another area of active research in VEs is in determining the effect of *update rates*, *refresh rates*, and *lag*. Traditionally, issues such as these arose from the lack of computing power needed to track users, evaluate models, and render images. Current advances in processor design and computer graphics help reduce lag and support the development of more complex, photorealistic scenes. "Lag" we loosely define to be the time between user action via some interface mechanisms (input devices) and perception of that action's results via user interface presentation components (output devices). Of particular interest is the manifestation of lag via visual output. [Richard et al., 1996] reports that delays of more than 300 milliseconds may decrease user presence and immersion. Display update and refresh rates can have a direct effect on visual lag, and as such, ***developers should strive for high refresh and update rates to minimize latency or lag*** ≪Visual8≫. For instance, a display's scene which is updated infrequently will increase user-perceived lag, as the effect of users' actions are not immediately realized. In such cases, high frame rates will not remedy lag, since the updated visual model is not made available for timely presentation. For example, assuming a dynamic environment, a display which has a frame rate of 60hz and refresh rate of 20hz will display a new model or view every three frames. If user actions are highly dynamic, actions performed within the three frames will not be realized until the next display update. Thus, it appears as if update rate is more crucial to usability than frame or refresh rate. Indeed, a 1994 study claims that low frame rates have minor effect on user performance of target acquisition tasks [Ware and Balakrishnan, 1994]. One should be cautioned though, as extremely low frame rates may produce "annoying illusory motion artifacts" which, when compounded with visual lag, may increase the likelihood of user sickness [Richard et al., 1996]. [Ware and Balakrishnan, 1994] offers the following suggestions when designing a VE involving 3D target acquisition tasks:

- **Use** ***input devices which have low lag*** ≪Visual9≫, ideally less than 50msec.
- If possible, **separate** ***head lag from hand lag*** ≪Visual10≫. This may be done by sampling the head-tracked device, drawing most of the scene, then sampling the hand-tracked device, and drawing the target and the cursor.
- If possible, **decouple** ***the target and cursor from the rest of the environment***

≪Visual11≫, so that higher update rates can be applied to the target and cursor only.

Types of VE Displays

Visual displays come in several different forms, including head-mounted displays, CAVEs™, counterbalanced displays, and virtual workbenches. A key question is "Why are there so many different types of displays?" A short answer may be because there are many different types of VEs involving many different types of user tasks. This then raises questions such as "Do tasks revolve around *individual* or *group* work?", "Are displays meant to convey an *environment* or a *model* ?", and "What degree of image quality or photrealism is desired?" It is unreasonable to assume that a single display device could support any and all VE tasks equally well. Instead it is obvious from the proliferation of display types that most display types are perfectly suited for some tasks, sufficient for some other tasks, and ill-suited, impossible, or intractable for others. Thus, to assess the effectiveness of a display device is to assess how well the device matches representative user tasks. In effect we can say that "tasks determine display" [Bennett et al., 1996].

What types of tasks are best suited for HMDs, SIDs, or VMDs?

One way to determine mappings from tasks to display is through the type of presence the tasks and system intend to convey. Consider the distinction between immersion, self presence, and object presence [Bennett et al., 1996]. Full immersion requires an enveloping display, so that all "external" (outside the VE) sights and sounds are omitted. Users become immersed in VE-generated information only. This is typically achieved through use of an HMD. Self presence is the perception that, from the user's perspective, "*I* am here." Immersion is not required to achieve self presence. ***Peripheral motion cues, location cues, and fov contribute to self presence*** as typically experienced through spatially immersive displays (SIDs) such as CAVEs™ and domes. Object presence, on the other hand, is not concerned with user or self presence. Instead it can be thought of as the degree to which users believe an object is present. Object presence is the perception that, from the user's perspective, "*it* is there." A ***good 3D perspective and head tracking are necessary for rich object presence***, typically provided through the use of virtual model displays (VMDs).

Given a particular VE, ***representative user tasks may implicitly suggest the mix of immersion, self presence, and object presence required*** ≪Visual12≫. In turn, these requirements may lend credence to a particular display type (see Figure 20). Of course there are other considerations when analyzing the appropriateness of a display type, including issues such as number of users, user mobility, and collaboration. These considerations are dicussed below, within the context of the three display types shown in Figure 20.

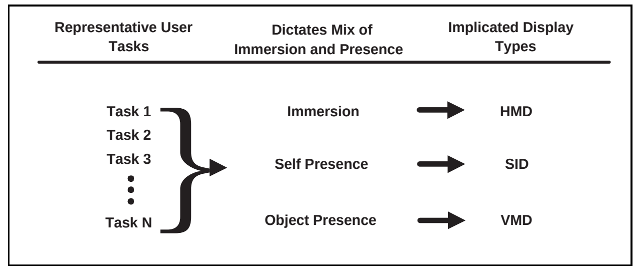

FIGURE 20: How Tasks May Determine Display Type [Bennett et al., 1996]

Head-Mounted and Other CRT-Based Displays

Perhaps the most well-known device in VEs is the head-mounted display. It was, after all, part of Ivan Sutherland's original conception of VEs [Sutherland, 1968]. Over the decades, advances in display technology have transformed the bulky, heavy, low resolution HMDs of the past into the lightweight, higher resolution, color HMDs of today.

HMDs are best-suited for single, autonomous user activity ≪Visual13≫ [Bennett et al., 1996]. Each user wears a separate display, which must provide a unique perspective depending upon user location, orientation, activity, and so on. In a multi-user setting, each HMD may need to also present all other users, with accurate location, orientation, and so on. Coordination of displays among a large number of users may be too computationally intensive, resulting in severe latency problems, and in effect, rendering the system useless. Tasks which require that multiple users occupy the same physical space are ill-suited for HMDs, as users contend for physical floor and room space without the ability to see each other.

On the other hand, scenarios involving several remote users may be better off using HMDs. In this fashion, users are able to occupy the same virtual space without having to rely on sharing the same physical space. Coordination of displays among users over a network *in real time* is not trivial. Researchers at the Naval Postgraduate School are developing strategies to address this problem, by attempting to minimize the bandwidth required to fully represent user activity.

Head-mounted displays allow users to move from "interface to inclusion" [Bricken, 1990]. In traditional systems, computer screens act as effective boundaries between users and information. HMDs allow users to move through this barrier, and become intimately included in the VE. Instead of looking *into* the fish tank, users are *in* the fish tank (the same could be said of fully immersive CAVEs™). As such, developers should strive for ***seamless user inclusion by eliminating obvious interface boundaries*** ≪Visual14≫.

A side effect of this inclusion, or immersion, is the fact that all "real-world" sights are eliminated so that users are left to experience the virtual world without distraction. This may be a useful feature in entertainment settings, where a goal is to isolate users within the gaming environment. Other application areas well-suited for complete immersion are those of architectural and interior design as well as walk-through simulation. In general, ***HMDs are well-suited for applications where complete visual immersion or absence of distractions is required*** ≪Visual15≫.

Traditionally, HMD field of view has been somewhat limited as compared to that of SIDs and VMDs. These displays usually provide 20 to 50 degrees of vertical field of view and 80 to 140 degrees of horizontal field of view [Kaiser Electro-Optics, 1996] [Fakespace Inc., 1996]. Although these displays are "immersive", it is rare that horizontal fov reaches that which we experience in the real world (i.e., the limit of human eyesight).

Field of View

As mentioned earlier, field of view and resolution are contradictory features. The problem is magnified in an HMD scheme, due to the limited number of pixels (as compared to SIDs and VMDs). Designers are faced with the dilemma of pixel allocation; the issue is whether to use a larger field of view and low resolution, or a smaller field of view and high resolution. This choice typically falls back onto some type of user task analysis;

that is, which of the two features is most appropriate for the user task(s) at hand. Another approach to this problem exploits a human's high visual acuity over small regions. This approach, developed at the University of North Carolina at Chapel Hill, uses a high-resolution insert against a low-resolution, wide field of view background [Yoshida et al., 1995]. By tracking users' eyes, the system is able to place and render the high-resolution insert in appropriate locations within the user's field of view. By limiting the size of the high-resolution insert, the display can allocate remaining pixels to maintain a wide field of view.

Physical Issues

Two of the more significant limitations of HMDs are their *comfort* and *mobility*. HMDs can weigh anywhere from less than a pound to more than five pounds, although most are in the two to three pound range [Kocain and Task, 1995]. When compared to shutter glasses and counterbalanced displays, HMDs are the heaviest of all. Thus, it is no surprise that user fatigue is associated with prolonged use of HMDs. HMDs are also designed to be a "one size fits all" device. And although manufacturers claim that their device will fit 95% of user heads, one has to wonder if the device fits 95% of users' heads *comfortably*. Compounding HMD weight is the fact that ***HMDs are typically tethered by audio and video cabling, limiting user mobility to cable length and support mechanisms*** ≪Visual16≫. In the absence of cable support arms, users must be "shadowed" by another individual who ensures that cable and users do not become intertwined. Even with a cable support system, users are aware that they are tethered and subsequently experience increased cognitive loads as they attempt to track their physical position with respect to tethering range. Given that the cabling system must attach to the user somewhere (or directly to the display), it has been suggested that the waist/small of back is a preferred location, minimizing interference with user movement.

The Fakespace binocular omni-oriented monitor (BOOM™), BOOM3C™, and PUSH™ Desktop Display (essentially a BOOM™ which sits on a desktop) reduce user fatigue associated with HMD size, weight, and comfort by providing a counter-balanced display into which a user looks. That is, users do not "wear" the display; instead the display is attached to a counter-balanced boom arm so that users may position the display in front of their eyes with little effort (see Figure 21). Because of this, the BOOM™ and PUSH™ displays are better considered members of the class of counterbalanced displays rather than HMDs. Perhaps one of the most appealing features of ***both the BOOM™ and PUSH™ Desktop Display*** is that they ***can be seamlessly integrated into user work activity, exploiting the work habits that users already have in place*** ≪Visual17≫ [Fakespace, 1997]. For example, engineers designing complex parts in a computer-aided design (CAD) environment may use the BOOM™ to get a three-dimensional, stereoscopic view of these parts without leaving their seat or putting on (and adjusting) an HMD. Once viewed, the engineer simply nudges the BOOM™ aside, and resumes work in the CAD environment. Moreover, BOOMs™ are easy to pass between people, so that sharing of information and perspective may be facilitated among colleagues working in a collaborative setting.

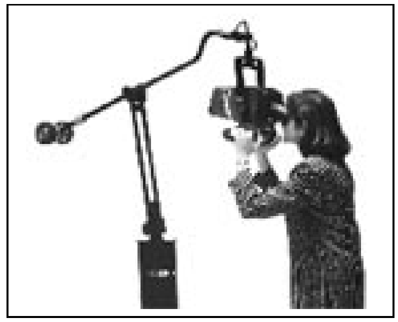

FIGURE 21: The Fakespace BOOM3C™ [Fakespace, 1997]

<u>See-through HMDs and Augmented Reality</u>

See-through HMDs, currently used mainly for augmented reality, allow users to view computer-generated images superimposed over the real world. These displays are typically integrated into existing work practices and procedures, providing real-time, supplemental information to users. For example, the University of North Carolina at Chapel Hill has developed an augmented reality system which allows surgeons to spatially visualize ultrasound data during breast biopsies [State et al., 1996]. Traditional procedures required doctors to position a needle inside the patient, while viewing the needle's ultrasound image on a CRT placed above the patient. This required excellent hand-eye coordination, since surgeons had to look in one direction (at the ultrasound monitor) while working with their hands in another direction (at patient). By rendering the ultrasound data via see-through

HMD at the spatially accurate position (e.g., on and within a breast), surgeons can look directly at the location at which the needle enters the patient.

Integration of an augmented reality system into Boeing's manufacturing process is a prime example of how a VE representation provides more useful, flexible, and accessible information to users [Boeing, 1996]. The assembly of large, complex aircraft cabling is traditionally done on sizable assembly boards containing positioning pegs and color-coded lines to aid assemblers in spatial aspects of construction. Since each cable assembly is unique (length, connectors, etc.), Boeing has to store a large number of these assembly boards, one for each unique cabling assembly. The augmented reality alternative, shown in Figure 22, superimposes positioning pegs and color-coded lines onto a generic peg board so that **all** cable assembly can be done on one board. Furthermore, the augmented reality system is able to superimpose additional assembly information at cable ends which is traditionally referenced in assembly manuals (such as detailed connector pin-outs). Presentation of this additional information saves assemblers the time involved in shifting their focus from the board in front of them to the assembly manual on a table nearby (or elsewhere). Time wasted looking up specific connector information is eliminated by the augmented reality system, as specific connector information is pre-programmed and presented at the appropriate time, namely, when the assembler is working on that particular connector.

FIGURE 22: Augmented Reality in Boeing's Manufacturing Process [Boeing, 1996]

Spatially Immersive Displays

Spatially Immersive Displays (SIDs) provide a balance between immersion and spatial object rendering by generating stereoscopic images on physical surfaces viewed by users through liquid crystal display shutter glasses. Typically the surfaces envelop the user to some degree, creating a sense of immersion. However, shutter glasses are necessarily transparent so that users see anyone or anything which may also be present inside and outside the computer-generated environment. The spatial quality of 3D images experienced by users of SIDs is far superior to that available through HMDs. Thus, ***SIDs are well-suited for spatially rich applications*** ≪Visual18≫ such as environmental walk-throughs and flight simulations.

<u>CAVEs™ and Domes</u>

The two most common examples of SIDs are CAVEs™ and domes. Images generated in a CAVE™ are presented on some combination of adjacent walls, floor, and ceiling of what can thought of as a simple room (see Figure 23 for a computer-generated image of a CAVE™ layout). By contrast, images in a dome display are projected onto the curved surface of the dome. By sheer magnitude of the display surfaces, both CAVEs™ and domes provide sufficient but not complete immersion. For example, in some CAVEs™ , images are projected only onto three walls and the floor.

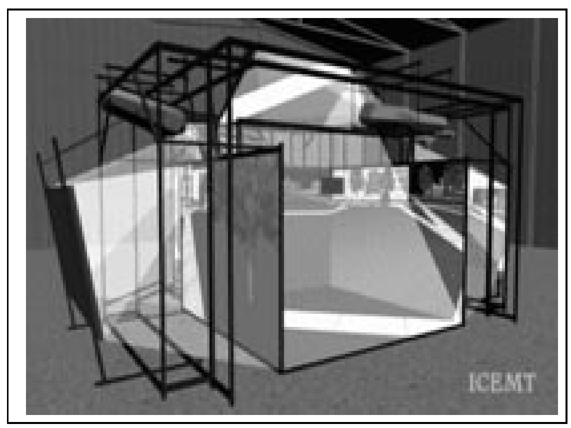

FIGURE 23: Iowa State's CAVE™ Configuration [http://www.icemt.iastate.edu/]

Thus, users may turn and look out the "back" (the fourth wall) or out the "top" (the ceiling) of the CAVE™ . This may be distracting or unsettling for some users, as they feel that the environment is incomplete. Perhaps the most significant implication of the missing wall is the restriction it places on user orientation. For example, consider the simple task of visual search. In the real world, and also in HMD-based VEs, users will stand in place and turn themselves around in a circular motion while visually scanning for a specific item. In a CAVE™ , the same type of task is typically performed in an intuitively inverted manner. Namely, by rotating the environment *around* *the user*, while physically facing a fixed direction (e.g., away from a missing wall) [National Center for Supercomputing Applications, 1996]. Rotation of the environment will eventually reveal the complete scene to users. However, the motion of the scene, as opposed to the user, may leave some feeling a bit queasy. A handful of CAVEs™ are including a fourth wall and a ceiling to fully immerse users. In these environments, rotation of terrain is not necessary, resulting in a more natural interactive interaction.

Advantages of SIDs

CAVEs™ and domes are generally considered to support enormous field of views ≪Visual19≫ when compared to HMDs. This provides users with great peripheral information which, in turn, enhances location and motion cues. Some contend that the vertical fov of CAVEs™ is limiting when ceiling imagery is not displayed. While this may be true, most tasks do not require users to constantly look up. Those tasks which do would most likely be extremely tiring.

What are the effects of not providing fourth wall and ceiling projections in a CAVE™ ?

The transparency of shutter glasses used in SIDs has its advantages as well. For example, users typically will have a good sense of self due to the subtle cues provided by their physical body. There is no need to present full-user computer-generated avatars; SIDs and shutter glasses afford the real thing. Although, typically SIDs (as well as HMDs and VMDs) will display some type of "cursor" such as a disembodied hand, crosshair, etc., to help users visualize their hand's current position for manipulation based tasks. Similarly, in a multi-user scenario, users are able to look about and see each other, providing each user with important information such as spatial relationships among user bodies and user activities. Thus, ***SIDs are typically considered well-suited for multi-user tasks and collaboration*** (as are VMDs) ≪Visual20≫. Adding another user to the environment costs no more than an additional pair of shutter glasses. Computation costs are constant, as no additional avatars and environment geometry are required.

Another advantage of SIDs revolves around the fact that users wear a simple, extremely lightweight pair of shutter glasses. The design is so light and unobtrusive that users may work in these environments for extended periods of time with minimal fatigue. Furthermore, the glasses (by themselves) are wireless so that users may be highly mobile without the distraction of bundled, tethering cables. Mobility is limited only by the size of the CAVE™ or dome. This is usually not a problem, as the types of environments generated to date in CAVEs™ and domes are typically user-centered, so that great degrees of mobility are not necessary.

Disadvantages of SIDs

Despite the apparent advantages of SIDs, there are of course several tradeoffs to consider. One of the most noted disadvantages of SIDs (and of VMDs) is related to stereoscopic image creation and user tracking. By design, shutter glasses require two separate images to be presented, one to each eye. The distance between the two images determines the 3D position of environment objects. Computations made to determine distance between images are based on a single viewer's or user's position. Since images are created on physical surfaces for all to "see", it is technically challenging and computationally expensive to generate a separate set of images per user. For example, given that images for a single user are presented at 60hz, supporting three separate user perspectives would require presentation at $60 \times 3 = 180$hz with no noticeable degradation in image quality. In most multi-user scenarios, users share a single set of images. Thus, from the perspective of the tracked user, the stereoscopic imagery is flawless. However, from the perspective of other, non-tracked users, the stereoscopic imagery will be skewed, or even disconcerting, depending upon the distance between non-tracked users and the tracked user. As such, ***SIDs are not well-suited for multi-user VEs that require separate images per user*** ≪Visual21≫. Domes avoid this problem by engineering large "sweet spots" in the room so that most of the viewers experience good stereoscopic imagery. CAVEs™ , however, have smaller "sweet spots" due to their diminutive size (when compared to theater-sized domes). A CAVE™ is typically 10' by 10' by 10', whereas domes can range from a personal, cockpit-sized display to an entertainment-based, theater-sized display. A low-tech remedy to this problem of multi-user tracking is to have users pass the tracker around when precise stereo image perception is crucial. For example, consider a VE museum tour where non-tracked users are effectively guided though an environment by a tracked user who, in turn, may stop and pass the tracker around at opportune moments.

Another limitation of most SIDs is their lack of mobility; CAVEs™ and domes are very non-portable. They are currently extremely expensive setups which are intended to be erected and remain in a single location. Thus research and use must revolve around the physical facility. A possible implication is that SIDs cannot be integrated in existing workplaces and activity as easily and inexpensively as can HMDs. Until technology can deliver paper-thin displays capable of being tiled, SIDs' real-world use and applicability may be somewhat limited.

Virtual Model Displays

Virtual Model Displays (VMDs) are a third class of display types providing three-dimensional visualization without complete immersion. In essence, VMDs are capable of generating virtual worlds where the effect is limited to the volume of space roughly equivalent to just inside and outside the display surface. The resulting lack of complete immersion is one of the major distinction between VMDs and SIDs. A limited form of immersion can be created by VMDs which have very large, upright display surfaces. Another distinction is the fact that, as the name states, ***virtual model displays are particularly well suited for providing exocentric views of virtual models*** ≪Visual22≫ such as a virtual patient [Naval Research Laboratory, 1997]. VMDs provide excellent *object presence*, supporting the notion that "it is there." These distinctions in turn suggest the types of applications and interactions best suited for VMDs.

FIGURE 24: EVL's ImmersaDesk™ [www.evl.uic.edu/EVL/]

Examples of VMDs include EVL's ImmersaDesk™ (see Figure 24), desktop or fishtank VE [Ware et al., 1993], and Wolfgang Krüger's Responsive Workbench [Krüger et al., 1995]. In all of these cases, the display is essentially a single presentation surface physically placed at some distance in front of the user. The major distinctions among specific instances of VMDs are size, dimension, and pitch or tilt of the display.

Characteristics of VMDs

The size and dimension of VMDs may have a direct effect on typical display characteristics such as resolution and field of view. In general, the larger the display, the greater the field of view, at the expense of resolution. Non-desktop VMDs such as the ImmersaDesk™ and the Responsive Workbench (shown in Figure 25) use rear-mounted RGB projectors to display computer generated images. Thus, as the display surface gets larger, the projected image gets less clear. But perhaps a more significant implication of a VMD's size and shape is effects on collaborative use. Much like SIDs, ***VMDs are well-suited for local collaboration, since multiple users can participate using the single display*** ≪Visual23≫. However, it is much easier for three users to stand around the Responsive Workbench, than it would be to crowd around a desktop monitor. Due to the physical footprint and near-vertical pitch of the display surface, users stand *in front* of the ImmersaDesk™ . The Workbench, on the other hand, has a near-horizontal pitch, allowing users to stand *around* the Workbench.

FIGURE 25: NRL's Responsive Workbench [www.ait.nrl.navy.mil/vrlab/pics_for_CGA.html]

The pitch or tilt of a VMD can effect usability in other manners as well. For example, display surface pitch has a direct impact on the "field-of-depth" the display affords. That is, the direction and extent of perceived depth *relative to the user* is a direct function of the display pitch [Bennett et al., 1996]. For example, consider the near-vertical pitch of the ImmersaDesk™ . Most of the depth available is along the axis *parallel* to the floor,

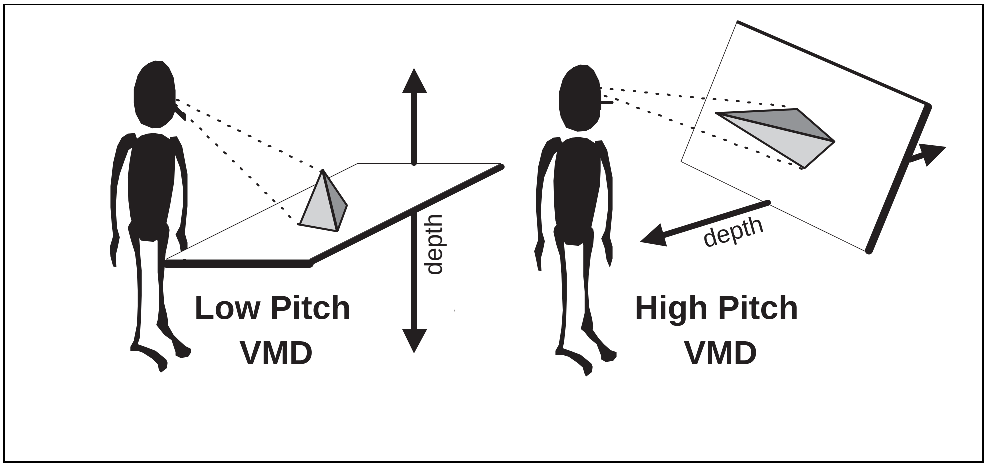

Figure 26: Perceived Depth as a Function of VMD Pitch [Bennett et al., 1996]

producing environments perceived as very deep, and somewhat immersive. On the other hand, the near-horizontal pitch of the Responsive Workbench creates depth along the axis *perpendicular* to the floor, producing environments perceived as short and squat. Figure 26 depicts a possible relationship between VMD pitch and perceived depth.

Thus, ***virtual model shape and size as well as desired point of view may suggest VMD pitch, and subsequently VMD type*** ≪Visual24≫. For example, near-horizontal pitch, such as that provided by the Responsive Workbench, are well-suited for exocentric manipulation of short and squat virtual models (e.g., virtual surgery). On the other hand, near-vertical pitch, such as that provided by the ImmersaDesk™, is well-suited for egocentric flight though deep models (e.g., architectural walk-through). Note that both the ImmersaDesk™ and Responsive Workbench display pitch are adjustable to some degree. Understanding the relationship between user, interactions, and virtual model shape may help designers choose an appropriate display pitch.

<u>Interaction with VMDs</u>

In some ways, user interaction with VMDs is similar to that of SIDs. For example, in both cases, users wear LCD shutter glasses during stereoscopic rendering. Thus, the same advantages associated with the transparency of shutter glass use in SIDs can be applied to

VMDs. Namely,

- users typically have a good sense of self due to the subtle cues provided by their physical body;
- there is no need for full-body computer-generated avatars;
- in a multi-user scenario, users are able to look about and see each other, providing each user with important information such as spatial relationships among user bodies and user activities;
- adding another user to the environment costs no more than an additional pair of shutter glasses; and
- computation costs are constant, as no additional avatars and environment geometry are required.

Since the basic metaphor in VMDs is the non-immersive virtual model, interaction with the model typically involves users moving about model images, as opposed to users being in among them. This suggests that VMDs may be better suited for exocentric point of views (as opposed to egocentric) and therefore, exocentric manipulation. As such, ***VMDs are particularly well-suited for model prototyping and other tasks which require manipulation of some external model*** ≪Visual25≫.

Navigation in VMDs is a little different, and as previously mentioned, may depend upon VMD pitch. For example, egocentric navigational schemes such as the "flight metaphor" may not be as usable in low-pitch VMDs since the pitch does not fully afford an egocentric point of view. In these cases, it may be more appropriate to simply treat the model as a generic object, reducing navigation to exocentric manipulation of an object. In high-pitch VMDs, an egocentric navigation scheme may be more plausible since a more egocentric point of view is available.

8.2.2 Aural Feedback — Acoustic Presentation

As we move through the world around us, we are bombarded by sound — sounds resulting from our own actions, sounds resulting from other's actions, natural and ambient sounds. As young minds develop, they learn to integrate sound into perception. Indeed, sound may be one of the major forces at play in perception. Studies have shown that ***aural feedback effectively improves user performance of tasks such as three-dimensional target acquisition and shape perception in single-user, desktop VEs*** ≪Aural1≫ [Mereu and Kazman, 1996] [Mereu, 1995] [Hollander and Furness, 1994] [DiGiano and Baecker, 1992]. Thus, as the push for more useful VEs ensues, researchers aim to develop more sophisticated virtual acoustic presentation.

Advantages of Aural Feedback

An advantage of acoustic presentation is increased user spatial awareness. [Mereu and Kazman, 1996] examines users' spatial abilities given visual and acoustic environments. Using a target acquisition task, researchers examined users' spatial awareness via distance estimation. Subjects aligned or docked a three-dimensional cursor to a point within a three-dimensional abstract object called a "blobby." Visually, subjects were presented with the abstract blobby and a cross-hair cursor. Cursor position was represented using tonal, musical, and orchestral sounds. The study showed that distance estimation via aural cues alone is very difficult. But when aural cues are used in conjunction with visual tasks, target errors were reduced 32 to 78% and task completion times were significantly lower than times for sound-only environment, which were 123 to 215% longer.

While the use of acoustic presentation in VEs appears helpful, it may not be necessary in all situations. As with other modes of communication, it is important to understand the difference between audio as necessarily inherent in functionality (voicemail, music browser, etc.) and audio as a complement to other sensory functionality. Given the temporal, non-persistent nature of audio, ***aural information must be presented in a meaningful, timely, and useful manner*** ≪Aural2≫. Based on a literature review, [Cohen and

Wenzel, 1995] lists some circumstances in which acoustic presentation is desired:

- when the *origin* of the message is itself a sound (voice, music)
- when other channels are *overburdened* (simultaneous presentation)
- when the message is *simple* and *short* (status report)
- when the message addresses *temporal* events ("Your process is finished")
- when *warnings* are sent, or when message prompts for *immediate* action ("Remote participant wishes to join group")
- when *continuously changing* (dynamic) information is presented (location, metric, or countdown)
- when *speech channels* are fully employed (virtual teleconferencing and collaboration)
- when a *verbal response* is required (compatibility of media)
- when *illumination* or *disability* limits use of vision (alarm clock)
- when the receiver *moves* from one place to another (employing sound as a ubiquitous I/O channel)

Some of these circumstances may occur simultaneously, giving even more weight to an acoustic presentation, such as in voice recognition interfaces. SRI International has developed such a system in which users interact via conversational dialog with a system agent named Simon [SRI, 1996]. Simon acknowledges command request and completion, so that a request such as "Simon, enable warning system" is acknowledged with a reply such as "warning system is enabled." Here the origin of the message is itself a sound (agent Simon), the message is simple and short, speech channels are fully employed, and the warning system's presence may not be observable.

Another advantage of acoustic presentation is that of ***directional content*** ≪Aural3≫. That is, a meaning can be associated with direction only, thus decreasing the cognitive load associated with listening and parsing a message whose meaning is conveyed in words. For example, in a VE-enhanced aircraft cockpit, pilots can be alerted to nearby aircraft by a tone emitted from the relative direction of the actual aircraft. Furthermore, the pitch, duration, and meter of the tone can be used to convey type of aircraft. This type of directional information is limited in the visual realm, since users cannot "see" in all directions at once.

A promising use of acoustic presentation involves ***empowering visually impaired users*** ≪Aural4≫. As a separate part of the Mereu and Kazman study, visually impaired subjects were tested using the same tasks and sets of acoustic cues [Mereu and Kazman, 1996]. Results showed that visually impaired users are quite capable of performing target acquisition tasks in 3D environments. Indeed, the visually impaired subjects outperformed their sighted counterparts in target accuracy when immersed in a sound-only environment! This is not surprising since sight impaired persons typically rely more heavily on, and thus further develop, their sense of hearing.

The development of acoustic presentation hardware has followed a path typical of other sensory devices. [Cohen and Wenzel, 1995] points out that the "evolution of I/O devices can be roughly grouped into generations that also correspond to number of dimensions."

Stereo Headsets and 3D Localized Sound

Stereophonic recording and presentation equipment have existed for decades. Yet just recently has audio of reasonable quality been integrated into computer systems and user interfaces. The introduction of headset audio into early VEs allowed developers to include acoustic information, without developing new, specialized equipment. More recently, the lure of headset audio may be based upon widespread availability and persuasiveness, giving VE developers a convenient, well-established, simple mechanism for acoustic presentation. Today, there are rich 3D spatially localized sound systems, designed to generate sounds which, from the user's perspective, appear to originate from any point within a 3D volume. These systems are capable of creating an aural space much like audio space we experience in the real world.

Perhaps the most limiting feature of current headset technology is the obvious fact that it can only define a two-dimensional space. Sounds originally sampled or modeled in three dimensions are presented through a left and right channel only. A shift in balance can create the illusion of sounds in between, but in the end, resulting sounds are diffuse, and limited to points between speakers [Cohen and Wenzel, 1995]. Thus, a major consequence of headset presentation is the lack of presentation "space."

One important role of acoustic "displays" is to organize sound information in some meaningful manner. Limiting the "space" in which to organize limits the possible organizational schemes, typically at the expense of usability. That is, a ***3D space allows for more complex sound separation, isolation, and position*** ≪Aural5≫ than that provided by simple headset audio. Moreover, ***three-dimensional space affords more detailed spatial and directional content*** than that provided by simple headset audio. On the other hand, if an application is better suited for a 2D sound space, such as a virtual teleconference where participants sit around a virtual table, use of a more complex 3D space may be unnecessary and confusing (i.e., voices propagating from a virtual ceiling could confuse a participant sitting at a virtual table).

Both headset and spatially localized 3D sound systems should ***strive to generate real-time sounds*** ≪Aural6≫, so that users' actions, observations, and experiences are further accentuated by timely aural information. As with real-time image rendering, acoustic refresh and update rates should be sufficient to avoid acoustic lag. The types and characteristics of sounds a system generates is another consideration. The number of different sounds, or range, a system can produce may be important in applications containing many different sound sources. However, if a system relies more on intensity levels, or volume levels, to distinguish among semantics, then one may be more concerned with the ability to generate observable differences in a single sound source.

Bandwidth

All the characteristics mentioned above may be fully utilized only if a sound system has sufficient bandwidth. VE system should ***provide high bandwidth aural channels to support simultaneous, dynamic presentation of many different sounds, from***

many different locations, at varying intensity levels ≪Aural7≫. When coupled with sufficient refresh and update rates, these high-bandwidth aural system should also be able to produce this complex sound space in real-time with correct respect to a moving user. For example, consider a user walking through a virtual ship. A high bandwidth system would be able to accurately reproduce the dynamic fabric of sounds one would encounter in such a walk. However, caution should be taken so as not to clutter users' sound space, since a major advantage in system-generated sounds is the ability to omit natural background noise, thus leaving the most important sounds unobstructed and clear.

Physical Issues

There are some unique usability considerations that 2D stereophonic *headsets* have which loudspeaker (2D or 3D) systems do not. For example, since a headset must be *worn* by users, its size, weight, and shape may have detrimental effects on user comfort, and consequently, usability. Furthermore, most headsets are hard-wired, so that users are effectively tethered. A loudspeaker system is not worn, so the tethering issue is of no concern. Aside from facility space limitations, loudspeaker size and weight have little effect on usability.

Perhaps the biggest limitation of a headset is its restricted applicability to multi-user VEs. In increasingly cooperative VEs, audio schemes that support multiple users at fixed cost (such as a 3D loudspeaker for a CAVE™ or dome) will be generally more useful than schemes requiring a *set of hardware* per user. Thus, ***loudspeaker audio is well-suited for for same-site, multi-user VEs*** ≪Aural8≫. The same can be said of *computational* complexity. As the number of users increases, the computing power required increases, as the system must support the increasing num-

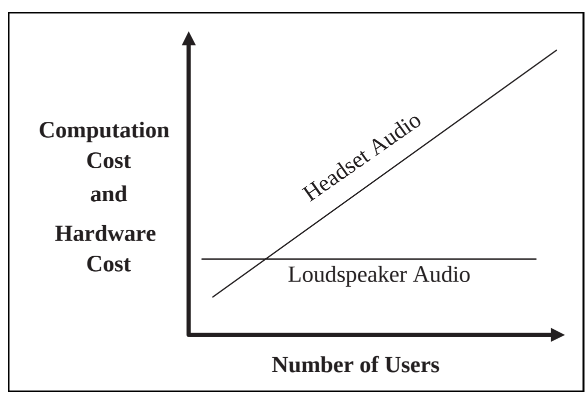

FIGURE 27: Number of Users vs. Audio Computational and Hardware Costs (adapted from [Bennett et al., 1996])

ber of audio headsets (each user may assume a different location within the VE). In a speaker system, one set of speakers provides the audio for as many users as the physical space allows. (Note that each sound is created in *real space* so that additional users hear the same sound, spatially accurate, regardless of position or orientation.) Figure 27 shows a possible effect of number of VE users on audio computational and hardware costs.

At this point one may ask, "When would one want to use a headset in a VE?" As previously mentioned, headsets offer a convenient, well-established, inexpensive, simple mechanism for acoustic presentation. If a VE includes a HMD, a headset may well be included as part of the HMD. Thus, cost is one of the most convincing arguments for 2D headset audio. Three-dimensional sound systems are extremely expensive. A good alternative may be a 2D speaker system, again, depending upon spatial needs of the application. Another ***benefit of headset acoustic presentation is that of portability*** ≪Aural9≫, in both the physical and compatible sense. When compared to loudspeakers, amplifiers, and processors, a headset is very lightweight. A VE with 3D sound should have no trouble presenting via headset, yet a VE modeled in 2D sound would not have enough spatial information to fill a 3D space.

8.2.3 Haptic Feedback — Force and Tactile Presentation

Many of the haptic devices present in today's VEs are based on research performed in the field of robotics. Indeed, an overwhelming amount of experimental results in the fields of robotics and VEs shows that the presence of haptic feedback increases task efficiency and accuracy in remote and/or virtual manipulation tasks [Shimoga, 1993] [Gomez et al., 1995] [Richard et al., 1996]. ***Haptic presentation is also effective in areas where other senses may not be usable*** ≪Haptic1≫, such as in acoustically muted or dark environments. For example, divers frequently make use of their sense of touch to navigate and explore surfaces while in muddy waters. An example from the manufacturing realm is that of remote inspection. Here harmful or lethal conditions once staffed by humans are occupied by robots, remotely "feeling" for part imperfections. Sensory substitution is yet another useful application area of haptics. The Braille system is an excellent example of how visually impaired persons are able to understand language communication (normally conveyed visually) via sense of touch.

An interesting note is that, unlike aural and visual senses, haptic sensations fundamentally involve a closely-coupled, bi-directional flow of information between users and environments [Richard et al., 1996]. That is, as we exert forces with our bodies we are simultaneously sensing reacting forces with our bodies. Thus, haptic hardware typically has to provide output while also supporting user-specified input. ***Other sensory information may be used to reinforce or enhance haptic tasks*** ≪Haptic2≫, typically resulting in improved user efficiency and presence. However, [Burdea and Coiffet, 1994] warns that too much redundancy may lead to sensory overload and disorientation.

Kinesthetic and Tactile Issues

VE users can be provided with a sense of touch and feel in two different ways: kinesthetic and tactile. Kinesthetic information is sensed through movement and/or force to muscles and joints. Tactile information, received through nerve receptors in the skin, conveys shapes and textures. [Kaczmarek and Bach-Y-Rita, 1995] contends that ***effective VE haptic displays need to include devices which provide both kinesthetic and tactile information*** ≪Haptic3≫.

Vibrations are an important part of tactile sensing and are key in performing everyday tasks. For example, vibrations help users recognize a physical state; the rattle of a screw indicates that it is loose, the vibrations generated when placing an object on a desk indicate object stability (or instability). Vibrations also help us refine docking tasks, such as placing a key in an ignition. Small vibrations upon initial contact provide cues such as force applied, approach angle, and even extent of grasp. While vibrations are present in many real-world tasks, it is interesting to note that most VEs use vibratory displays haphazardly. [Kontarinis and Howe, 1995] suggest that ***VEs need to use vibratory cues as they inherently exist in real-world tasks, and not simply as generic tactile cues*** ≪Haptic4≫. They further point out that these vibrations typically occur in the aural range (e.g., one normally feels **and** hears the rattle of a screw). Thus, vibratory information is easily generated by standard audio output, reducing cost and supporting tightly-coupled multimodal feedback.

As mentioned above, it is generally accepted that VEs should provide both kinesthetic and tactile displays. This is duly justified given that humans use feedback of both types, synergistically, in almost every task. [McNeely, 1993] offers the example of changing an automobile's sparkplugs. Providing tactile, aural, and visual feedback alone is insufficient, as the human body tires easily without proper support. The simple force feedback provided by the auto's body and engine is just as important as the other feedback channels. Kinesthetic and tactile displays need not be supported separately. In fact, it is possible for tactile technology to act as a surrogate for force feedback [Kaczmarek and Bach-Y-Rita, 1995]. For example, force feedback over a large area may be accomplished using the simultaneous activation of many small tactile presentation devices, or tactors.

While kinesthetic and tactile sensing typically work together, it is best to keep the various types of presentation cues separate ≪Haptic5≫ [Kaczmarek and Bach-Y-Rita, 1995]. That is, if a vibratory approach to displaying tactile information is used (which it typically is), it should not also be used to convey kinesthetic information within a given VE application or session. Mechanisms which provide force and pressure are better suited for kinesthetic communication.

Advantages of Haptic Devices

Some desired characteristics of haptic devices can be found in current VE literature. For example, during a 1995 *VRAIS* panel, Steve Jacobsen, of the University of Utah, presented the following characteristics as important for dexterous haptic devices [McNeely et al., 1995]:

- ***High bandwidth force reflection with high stiffness between master and slave devices*** ≪Haptic6≫;
- ***Strength and speed for natural end use*** ≪Haptic7≫;

- ***High resolution force and position presented to users*** ≪Haptic8≫;
- ***Reliable, intuitive, low fatigue operation*** ≪Haptic9≫.

In the 1995 *VRAIS* panel, Jacobsen also addressed the technology needed to exploit these desired characteristics.

Bandwidth appears to be a commonly noted characteristic of haptic devices. The need for high bandwidth results from the wide range of frequencies generated by the interaction between real-world forces and objects. Thus, a device capable of high bandwidth haptic presentation will be able to simulate a large number of shapes, textures, forces, and interactions (e.g., grasping, moving, deforming). As mentioned earlier, tactile vibrations associated with everyday tasks typically occur at relatively high frequencies (e.g., detecting a loose screw, identifying a dirty ball-bearing, texture recognition). Most kinesthetic forces are comparatively low [Brooks et al., 1990]. Moreover, contact between objects and hard surfaces generates a wide range of frequency components [Hannaford and Venema, 1995]. Thus, if devices are to provide high-performance haptic feedback, they will certainly need to ***support high bandwidth haptic interaction*** ≪Haptic10≫.

Kenneth Salisbury, also a member of the 1995 *VRAIS* panel, warned that although good bandwidth and resolution are desired, we should ***be wary of complex, multi-degree-of-freedom systems*** ≪Haptic11≫ [McNeely et al., 1995]. Such systems tend to be very expensive and difficult to use. Furthermore, there are problems associated with the weight, force-power, and safety of such devices. Sensable's Phantom haptic device developed at the Massachusetts Institute of Technology (MIT), on the other hand, represents a well-designed balance of these concerns. Users insert a finger into a thimble attached to a mechanical linkage. The linkage in turn provides kinesthetic feedback through use of motors at the base of the linkage. Figure 28 shows Sensable's latest version, the Phantom3,

FIGURE 28: Sensable's Phantom3 Haptic Device [SensAble Technologies, Inc., 1997]

streamlining the base and offering more movement through longer linkage. Using two phantoms, VEs can model much more complex interactions, as users are able to "touch" and "feel" using finger and thumb. The Phantom is capable of simulating complex user interactions with a wide *range* of virtual objects, shapes, and textures even though it restricts movement to three degrees of freedom.

<u>Spatial Resolution</u>

Another important characteristic of haptic devices is spatial resolution. When assessing the spatial resolution of tactile displays or tactile display content, it is important to note that human skin has limited spatial resolution capabilities [Kaczmarek and Bach-Y-Rita, 1995]. One implication of this fact is that ***tactile displays need not provide incredibly high resolution*** ≪Haptic12≫. The CyberGlove™ from Virtual Technologies is a prime example, as it uses a single vibrating element on each finger. Higher resolution simply drives up the cost, with possibly no greater impact on usability.

Another implication of our skin's limited spatial resolution is that ***simultaneous presentation of complex patterns, sensations, or objects should be avoided*** ≪Haptic13≫. Instead, VE designers should consider presenting simplified versions of objects, generated by tracing or edge enhancement. As with other modes of sense, only important, distinguishing features of objects typically should be presented. Similarly, ***haptic presentation should be cautious in presenting, and semantically binding a large number of intensity levels*** ≪Haptic14≫ [Kaczmarek and Bach-Y-Rita, 1995]. Visually analogous to gray levels, there is a point at which users either cannot distinguish between intensity levels, or cannot make the cognitive association between intensity level and meaning.

What degree of tactile spatial resolution effects usability?

Categories of Haptic Devices

Haptics fall behind other areas of VE development. William McNeely, chair of a 1995 *VRAIS* panel on force feedback, suggests that this is possibly due to the lack of "commercial infrastructure" enjoyed by visual and aural fields [McNeely et al., 1995]. Nevertheless, there have been some ground-breaking developments in the quest to provide the sense of touch to VE users. Haptic devices are typically implemented via mechanical, electrotactile, and/or vibrotactile means.

McNeely classifies haptic displays into three categories:

1. *User-worn*;
2. *Hand-held*; and
3. *Encountered*.

User-worn Devices

While user-worn devices encompass a wide range of equipment including gloves, exoskeletons, and body suits, it is typically the glove which receives the most attention. Haptic gloves apply to a small part of the user's body, yet are perhaps the most useful. Most tasks done in the real world are performed by the hands. Likewise, many of the tasks performed in VEs involve manipulating virtual objects with hands.

User-worn haptic devices, much like other user interface devices, must be designed so as to ***maximize user comfort and ease of use*** ≪Haptic15≫. [Wiker et al., 1991] show that user fatigue and discomfort are aggravated by factors such as grasping force and work-to-rest ratio. If the force feedback is too strong, or disproportional to the task at hand, then users' grasping muscles will tire. The result is a user who is less capable of correctly estimating force magnitudes and variations. This, in turn, may trigger increased task error and performance times.

Hand-worn devices should be lightweight, unencumbering, and portable enough to allow users sufficient freedom of motion ≪Haptic16≫, yet powerful enough to simulate real-world forces. Moreover, the device should ***allow uninhibited, effortless motion when no virtual forces or contact are at play*** ≪Haptic17≫ [Hannaford and Venema, 1995]. That is, a user's hand should effortlessly close when no object is being held.

An example of such a portable, lightweight, force feedback device is the Rutgers Master II (see Figure 29). Equipped with pneumatic actuators, the Rutgers Master II is able to provide precise force feedback to each individual finger. Tests with the original Rutgers Master reveal that virtual manipulation task performance accuracy may be increased up to 50%. Learning time may also be reduced up to 50% when compared to environments providing no force feedback [Richard et al., 1996].

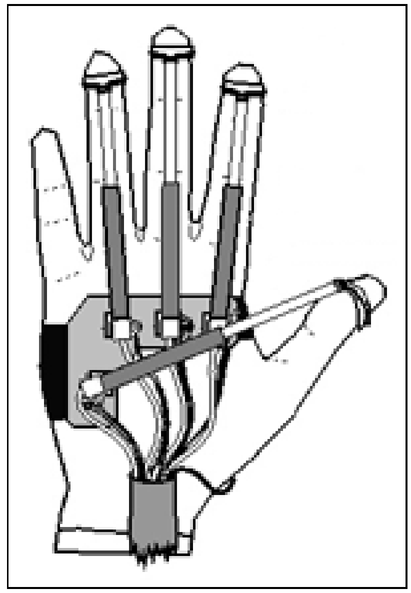

FIGURE 29: The Rutgers Master II [Burdea et al., 1992]

Another type of haptic glove available to VE developers uses small vibrating pin units in the fingers and palms in place of mechanical linkage and actuators to provide tactile feedback to users. These gloves afford a bit more mobility than force feedback gloves at the expense of kinesthetic presentation. Many times, vibratory feedback is used to indicate that some object is being touched or held. Developers should be wary not to use this type of tactile feedback glove for tasks which would not have such feedback in the real world. That is, just because a glove *can* provide vibratory feedback does not mean that such feedback is *appropriate*. Simply put, tactile feedback is most appropriate for tasks inherently involving tactile feedback.

On the other hand, mechanical exoskeletons are typically heavy, bulky, and uncomfortable to wear. Because they are worn (usually covering chest, arms and sometimes legs), there is a concern for size and fit. Another issue is whether the ***joints of the device fit the joints of the user so that body movement is natural and not forced*** ≪Haptic18≫ [Zyda et al., 1995]. Weight and fit (along with other factors) combine to affect user mobility, which is easily restricted and short-lived when not optimized.

As mentioned earlier, the bandwidth required to present high-fidelity haptic information is quite large. Indeed, haptic information can range from high frequency tactile to low frequency kinesthetic. Most haptic gloves are unable to display both low and high frequency information. For example a force feedback glove is designed to provide low frequency, kinesthetic information, while a tactile glove is designed to provide a high frequency sense of touch. Given a choice of the two, designers may ***base haptic device design or purchasing decisions on the nature and frequency of representative user tasks*** ≪Haptic19≫. [Shimoga, 1993] presents an interesting diagram describing important milestones along the continuum of human finger bandwidth. This compilation of findings indicates the minimum bandwidth required for users to

- *react* to unexpected force,
- *apply force* and *motion commands* comfortably, and
- be able to *meaningful perceive* haptic information.

It also indicates the maximum bandwidth beyond which users cannot sense vibrations during skillful manipulation, distinguish between two consecutive force signals, and correct their grasping forces for slipping objects. Furthermore, this information may be useful in analyzing choice of haptic display by facilitating mappings between user task specifications and bandwidth. The bandwidth (along with other criteria) may then, in turn, suggest a particular haptic device.

Hand-held Devices

Force feedback joysticks, desktop manipulators, and specialized medical force feedback devices are all instances of hand-held haptic devices. The robotics community is to be credited with much of the development in this field, and as such, most hand-held devices deliver force feedback kinesthetic information and not tactile information.

Some desktop manipulators have been developed to present both kinesthetic and tactile information to users [Kontarinis and Howe, 1995]. Studies performed with these devices show that ***users perform manipulation-based tasks***, such as tactile inspection of machined ball bearings, puncturing a thin membrane, and placing pegs in tight holes, ***with significantly less error when either kinesthetic or tactile information is provided*** ≪Haptic20≫ when compared to the absence of any haptic feedback. In the case of the ball bearing test, subjects presented with tactile and visual information were able to select the bad ball bearing 66% of the time, as compared to 53% for the visual only trials. Furthermore, the inclusion of both kinesthetic and tactile information decreased task error rates (90% of the time the correct ball bearing was selected) even beyond those of the singleton trials.

Joysticks and Specialized Haptic Devices

Joysticks are a popular and highly accessible means of providing force feedback to VE users. Unfortunately, they do not naturally fit most tasks and in fact may draw users' attention away from the task and to the device. That is, the ***interaction between VE and user may be hampered by the fact that users manipulate* external *devices*** ≪Haptic21≫. For example, consider the task of selecting some virtual object. Positioning an avatar with a joystick is not as natural or intuitive as simply reaching out with a tracked glove. Another limitation of force feedback joysticks is the relatively small working volume imposed upon the user and subsequently the task. Moreover, such devices typically cannot provide force feedback to individual fingers [Gomez et al., 1995].

Specialized haptic devices, such as those used in surgical simulation (e.g., endoscopic surgery) [George Washington University, 1996][High Techsplanations Inc., 1996] are typically expensive to develop and have extremely limited applicability. However, for the purpose of realistic, specialized training such as surgery, for example, they represent a viable solution.

Both ***joysticks and specialized devices*** suffer from their lack of mobility, usually due to weight or size. Thus, these devices ***are better suited for relatively stationary tasks*** ≪Haptic22≫, that is, tasks in which users do not move about. For example, technicians remotely inspecting small parts for defects would sit at a station and "work" in a small volume directly in front of them.

Tasks which require a high degree of resolution and precision, such as micro-surgery, require devices of high bandwidth. Pen-based force displays typically meet this requirement, providing the high resolution, low inertia, and low friction interaction needed to accurately model a surgeon's use of a scalpel [Buttolo and Hannaford, 1995].

The University of Virginia has been developing an insightful and elegantly simple technique of integrating hand-held haptic devices into VEs [Hinckley et al., 1994a] that frees users of fixed working volumes. Termed "tools with mass," researchers attach trackers to real-world tools normally associated with the given task. The "tools" effectively ***provide***

users with the same natural, gravitational, and inertial kinesthetic feedback in the VE as they experience in the real world ≪Haptic23≫. Studies have shown that these real-world props afford decreased learning time and increased usability when compared to virtual renderings of tools manipulated with a glove.

Encountered Devices

Encountered devices are those devices which provide haptic feedback simply by physically existing in the real world. That is, physically encountered devices are placed in the VE usage space, waiting for users to encounter and interact with them. Device positions are known by the system and are part of the virtual model. The simplest example of such a device could be a desktop. In a virtual office, the virtual desk is represented physically by a real desk. Users can work at, lean on, pound on, and even sit upon the desk with the most realistic kinetic and tactile feedback possible. Of course, this approach is highly limited and even somewhat arcane.

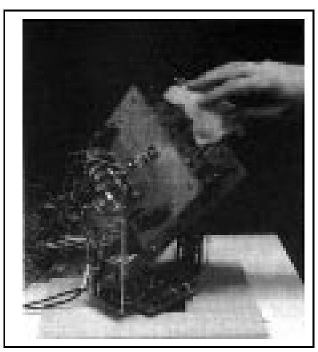

FIGURE 30: Early Surface Display Prototype [Hirota and Hirose, 1995]

Another approach, known as *surface displays*, makes use of robot technology to generate surfaces and objects. In surface displays, the idea is to present the surface of a virtual object itself, rather than the sensation of an object's force and/or feel (see Figure 30) [Hirota and Hirose, 1995]. The surface display is capable of rendering the feel of tools as well. Thus, a single display affords the benefits of force feedback and tactile feedback applicable to a large number of virtual tools.

Perhaps one of the biggest advantages of surface displays is that contact between user and object is made in the real world. Thus, the contact "feels real" because it is real. Moreover, the surface display itself detects user contact, or collision, a job typically done mathematically in software. This arrangement allows for accurate measurement of applied user force, which in turn is supplied to the simulation. Another advantage of surface displays is that users wear no additional equipment; interaction is extremely natural as no hardware intervenes between user and "objects."

Despite the advantages unique to surface displays, inherent limitations exist which restrict widespread use and acceptance. Foremost is the limited range of displayable objects and surfaces. This is due in part to the nature of the mechanical design; that is, surfaces are generated by a grid of protruding rods of varying length. For instance, an entire spherical object cannot be displayed such that users are able to wrap their hand and fingers around it. Surface and object rendering is also limited by the available presentation area. That is, a virtual surface larger than the display surface must be presented in parts. While this may not have an effect on single user tasks, it would certainly affect the usability of environments in which more than one user examines a common surface. Another concern centers around the use of mechanical parallel rods. The display's spatial resolution is typically low, as it is defined by the diameter of rods used. Given the limited resolution of human skin, the limited spatial resolution of surface displays may suitable for some tasks.

8.2.4 Environmental Feedback and Other Presentations

Visual, haptic, and aural presentation are well-established in today's VEs. However, other sensory information, such as olfactory, is typically ignored. If VEs are to create a strong sense of presence, they will need to ***provide additional sensory information other than the "big three"*** ≪MiscCues1≫ [Carter, 1992]. The lack of VE-generated olfactory information may be detrimental to usability, as ambient scents are inevitably present in any real-world setting. These ambient scents may provide conflicting olfactory cues with respect to other VE-generated sensory information [Barfield and Danis, 1996]. For example, the industrial scent of a laboratory would decrease presence for soldiers training for a smoke-filled battlefield environment.

The use of olfactory information in VEs could offer new application contexts such as training for the identification of hazardous chemicals. Other contexts include navigational cues by scent, such as room/area identification and path recognition (i.e., "trailing" a scent) [Carter, 1992]. Barfield also adds that much like aural information, ***olfactory in-***

formation can be used to provide directional and distance cues ≪MiscCues2≫. For example, the scent of smoke provides directional and distance information about a fire [Barfield and Danis, 1996].

Unfortunately, relatively little work has been done in olfactory presentation for VEs. This is likely due to inherent difficulties associated with producing and controlling smells. Current scent generation, such as perfumes and household air-fresheners, only produce single and often overwhelming scents. An olfactory display for VEs should be able to generate a wide range of scents. Another problem is that of scent dispersion, or controlling the scent's concentration, range, and duration. Consider a VE user moving through an environment. An olfactory component would have to be highly dynamic, producing scents at various intensities at various times. [Barfield and Danis, 1996] give an excellent discussion of the difficulties associated olfactory uses in VEs.

What is the minimal effective use of olfactory and other environmental features?

Other presentations that may increase usability include what we term "environmental presentations." These include characteristics such as temperature, humidity, and wind. In some situations, the existence of such characteristics may have a drastic effect on presence and usability. For example, to properly train personnel to perform a specific task in a hot, wind-fueled sandstorm, ***temperature and wind entities (and their effects) should be presented in full force to increase the believability, usability, and training transfer associated with the system*** ≪MiscCues3≫.

Temperature, humidity, wind velocity, and wind direction present themselves through a number of senses and objects. In the sandstorm example, wind velocity and direction are manifest visually as blowing sand in a particular direction, haptically as the feel of blowing sand against the body with a particular force, and aurally as wind howling through grains of sand. Temperature, although directly perceived though haptic senses, is manifest in other objects, since high temperatures may cause objects to feel or smell differently than they would under cooler conditions. Given the complexity of interactions among environmental presentations and other VE objects, accurate inclusion of complex environmental presentations in VEs may be a formidable task. The inclusion of simplified environmental presentations is then recommended for VEs in which such environmental entities play a major role, such as in the sandstorm example. Here importance is laid on the ability to perform a task under conditions of extreme heat and low visibility, two conditions which can be easily generated using current technology.

9 From Here to Where — Future Work

One of the first items on our future agenda is to develop a full WWW implementation of the taxonomy. The non-linear nature of hypermedia is well-suited for the taxonomy. In particular, we plan to exploit the use of hyperlinks to provide a more "usable" and "navigable" thesis. Links to other resources will also be included, such as links to academic, commercial, and government VE research labs. We will also provide direct links to specific VE products and applications mentioned in the taxonomy, and from cited literature to appropriate and available online papers and articles.

Although our current taxonomy organization, based on the theory of activity as discussed in Section 3, imposes certain interrelationships within the usability space, it may not be the most useful ordering for a particular "taxonomy user". We are currently considering including in the Web version dynamic ordering and filtering based on the needs of individual taxonomy users. For example, if an interested developer is researching usability issues of display devices, a re-ordered taxonomy could be generated which structures and ranks both explicit and implicit display issues.

Another benefit of a web-based implementation is the widespread availability the web provides. Once available, we expect interested parties to "use" the taxonomy and provide feedback to aid in the constant process of updating and refining the taxonomy.

We also hope to integrate and structure the taxonomy using Donald A. Norman's highly regarded "theory of action," which defines several *stages of activity* and interdependencies among the phases inherent in any interaction between human and machine [Norman, 1990]. We find this framework to be particularly well-suited for addressing how individual usability issues fit into a more abstract, larger scale understanding of interaction between users and VEs. A more complete discussion of the taxonomy as it relates to Norman's theory of action will be developed in subsequent research.

This taxonomy will also serve as a foundation upon which development of new usability engineering methods for VEs can be based. Through iterative development, we hope to refine a set of high-impact usability engineering methods specifically for VEs. Once developed, these methods in turn may be integrated into the overall system development lifecycle, creating better VEs which are less expensive to maintain, support, and use. The methods may also be used to evaluate existing VE applications, providing more "user-oriented" requirements in subsequent releases.

References

Alusi, G., Tan, A. C., Linney, A. D., Raoof, K., and Wright, A. (1997). Three dimensional tracking with ultrasound for augmented reality applications in skull base surgery. In *CVRMed-MRCAS '97. First Joint Conference — Proceedings of Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery*, pages 511–517.

Applewhite, H. (1991). Position tracking in virtual reality. In *Proceedings of Virtual Reality '93. Beyond the Vision: The Technology, Research, and Business of Virtual Reality*, pages 1–8, Westport, CT.

Ascension Technology Corporation (1997). Burlington, VT, USA (http://www.ascension-tech.com/).

Badler, N., Manoochehri, K., and Baraff, D. (1986). Multi-dimensional input techniques and articulated figure positioning by multiple constraints. In *Proceedings of the 1986 ACM Workshop on Interactive 3D Graphics*, pages 151–170.

Barfield, W. and Danis, E. (1996). Comments on the use of olfactory displays for virtual environments. *Presence: Teleoperators and Virtual Environments*, 5(1):109–121.

Barfield, W., Hendrix, C., and Bystrom, K. (1997). Visualizing the structure of virtual objects using head tracked stereoscopic displays. In *1997 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 114–119.

Barfield, W., Zeltzer, D., Sheridan, T., and Slater, M. (1995). Presence and performance within virtual environments. In *Virtual Environments and Advanced Interface Design*, chapter 12, pages 473–513. Oxford University Press.

Benford, S. (1996). Shared spaces: Transportation, artificiality, and spatiality. In *Computer-Supported Cooperative Work (CSCW '96) Conference Proceedings*, pages 77–86.

Benford, S., Bowers, J., Fahlen, L. E., Greenhalgh, C., and Snowdon, D. (1995). User embodiment in collaborative virtual environments. In *Human Factors in Computing Systems, CHI '95 Conference Proceedings*, pages 242–249.

Bennett, D., Chapelle, B. D. L., Zeltzer, D., Bryson, S. T., and Bolas, M. (1996). Information from the *SIGGRAPH '96* Panel Session, "The Future of Virtual Reality: Head Mounted Displays versus Spatially Immersive Displays".

Boeing (1996). Information collected during a visit to Boeing, Seattle, WA, on July 25, 1996.

Bowers, J., Pycock, J., and O'Brien, J. (1996). Talk and embodiment in collaborative virtual environments. In *Human Factors in Computing Systems, CHI '96 Conference Proceedings*, pages 58–66.

Bowman, D. A., Hodges, L. F., and Bolter, J. (1996). The virtual venue: User-computer interaction in information-rich virtual environments. Technical Report GIT-GVU-96-22, Graphics, Visualization, and Usability Center, Georgia Tech.

Bowman, D. A., Koller, D., and Hodges, L. F. (1997). Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. In *1997 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 45–52.

Bricken, M. (1990). Virtual worlds: No interface to design. Technical Report R-90-2, HITLab, University of Washington. Technical report available online at http://www.hitl.washington.edu/publications/r-90-2/.

Bröckl-Fox, U., Kettner, L., Klingert, A., and Kobbelt, L. (1994). Using three-dimensional hand-gesture recognition as a new 3D input technique. In *Artificial Life and Virtual Reality*, chapter 12, pages 173–187. John Wiley & Sons, Inc.

Broll, W. (1995). Interacting in distributed collaborative virtual environments. In *1995 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 148–155.

Brooks, Jr., F. P., Airey, J., Alspaugn, J., Bell, A., Brown, R., Hill, C., Nimscheck, U., Rheingans, P., Rohlf, J., Smith, D., Turner, D., Varshney, A., Wang, Y., Weber, H., and Yuan, X. (1992). Six generations of building walkthrough: Final technical report to the National Science Foundation. Technical Report TR92-026, UNC-CH Department of Computer Science (http://www.cs.unc.edu/).

Brooks, Jr., F. P., Ouh-Young, M., Batter, J. J., and Kilpatrick, P. J. (1990). Project grope — haptic displays for scientific visualization. *Computer Graphics*, 24(4):177–185.

Burdea, G. and Coiffet, P. (1994). *Virtual Reality Technology*. John Wiley & Sons, Inc.

Burdea, G., Langrana, N., Roskos, E., Silver, D., and Zhuang, J. (1992). A portable dexterous master with force-feedback. *Presence: Teleoperators and Virtual Environments*, 1(1):18–28.

Buttolo, P. and Hannaford, B. (1995). Pen-based force display for precision manipulation in virtual environments. In *1995 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 217–224.

Buxton, W. (1983). Lexical and pragmatic considerations of input structures. *Computer Graphics*, 17(1):31–37.

Card, S. A., Mackinlay, J. D., and Robertson, G. G. (1990). The design space of input devices. In *Human Factors in Computing Systems, CHI '90 Conference Proceedings*, pages 117–124.

Card, S. A., Mackinlay, J. D., and Robertson, G. G. (1991). A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems*, 9(2):99–122.

Carlsson, C. and Jää-Aro, K.-M. (1995). The DIVE laboratory. *Presence: Teleoperators and Virtual Environments*, 4(4):431–440.

Carter, J. P. (1992). The nose have it! *Presence: Teleoperators and Virtual Environments*, 1(4):493–494.

Cohen, M. and Wenzel, E. M. (1995). The design of multidimensional sound interfaces. In *Virtual Environments and Advanced Interface Design*, chapter 8, pages 291–346. Oxford University Press.

Cohen, P. (1992). The role of natural language in a multimodal interface. In *Proceedings of ACM Symposium on User Interface Software & Technology (UIST '92)*, pages 143–149.

Darken, R. P. (1996). Characterization of mobility platforms for interfacing humans to virtual environments. Department of Electrical Engineering and Computer Science. Sc.D. Dissertation. George Washington University. Washington, DC.

Darken, R. P. (1997). Personal Communication.

Darken, R. P. and Sibert, J. L. (1995). Navigating large virtual spaces. *International Journal of Human-Computer Interaction*.

Darken, R. P. and Sibert, J. L. (1996). Wayfinding strategies and behaviors in large virtual environments. In *Human Factors in Computing Systems, CHI '96 Conference Proceedings*, pages 142–149.

Davies, C. (1996). Information from the *SIGGRAPH '96* Panel Session, "Cognition, Perception, and Experience in the Virtual Environment: Do You See What I See?".

Davis, E. T. and Hodges, L. F. (1995). Human stereopsis, fusion, and stereoscopic virtual environments. In *Virtual Environments and Advanced Interface Design*, chapter 5, pages 145–174. Oxford University Press.

DiGiano, C. J. and Baecker, R. M. (1992). Program auralization: Sound enhancements to the programming environment. In *Proc. Graphics Interface '92*, pages 44–52.

Drascic, D. (1991). Skill acquisition and task performance in teleoperation using monoscopic and stereoscopic video remote viewing. In *Proceedings of the Human Factors Society 35th Annual Meeting*, pages 1367–1371.

Egan, D. E. (1988). Individual differences in human-computer interaction. In Helander, M., editor, *Handbook of Human-Computer Interaction*, pages 543–568.

Esposito, C. (1996). User interfaces for virtual reality systems. In *Human Factors in Computing Systems, CHI '96 Conference Tutorial Notes*.

Exos (1997). Woburn, MA, USA.

Fairchild, K. M., Lee, B. H., Loo, J., Ng, H., and Serra, L. (1993). The heaven and earth virtual reality: Design applications for novice users. In *1993 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 47–54.

Fakespace (1997). Menlo Park, CA, USA (http://www.fakespace.com/).

Fakespace Inc. (1996). Information collected during a visit to Fakespace Inc., Menlo Park, CA, in July, 1996.

Fleishman, E. A. and Quaintance, M. K. (1984). *Taxonomies of Human Performance*. Academic Press.

Foley, J. D., Wallace, V. L., and Chan, P. (1984). The human factors of computer graphics interaction techniques. *IEEE Computer Graphics and Applications*, 4(11):13–48.

Fukumoto, M., Mase, K., and Suenaga, Y. (1992). Finger-pointer: a glove free interface. In *Human Factors in Computing Systems, CHI '95 Conference Poster Presentation*.

George Washington University (1996). Information collected during a visit to George Washington University, Washington, D.C., on June 26, 1996.

Gibson, J. J. (1986). *The Ecological Approach to Visual Perception*. Hillsdale, NJ: Lawrence Erlbaum.

Gomez, D., Burdea, G., and Langrana, N. (1995). Integration of the Rutgers Master II in a virtual reality simulation. In *1995 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 198–202.

Greenhalgh, C. (1997). Massive — model, architecture and system for spatial interaction in virtual environments. Document available online at http://www.crg.cs.nott.ac.uk/ cmg/massive.html/.

Greenhalgh, C. and Benford, S. (1995). Massive: A collaborative virtual environment for teleconferencing. *ACM Transactions on Computer-Human Interaction*, 2(3):239–261.

Guiard, Y. (1987). Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *The Journal of Motor Behavior*, pages 486–517.

Hannaford, B. and Venema, S. (1995). Kinesthetic displays for remote and virtual environments. In *Virtual Environments and Advanced Interface Design*, chapter 10, pages 415–436. Oxford University Press.

Harmon, R., Patterson, W., Ribarsky, W., and Bolter, J. (1996). The virtual annotation system. In *1996 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 239–245.

Hauptmann, A. G. (1989). Speech and gestures for graphic image manipulation. In *Human Factors in Computing Systems, CHI '89 Conference Proceedings*, pages 241–245.

High Techsplanations Inc. (1996). Information collected during a visit to High Techsplanations Inc., Rockville, MD, in June, 1996.

Hinckley, K., Pausch, R., Goble, J. C., and Kassell, N. F. (1994a). Design hints for spatial input. In *Proceedings of ACM Symposium on User Interface Software & Technology (UIST '94)*, pages 213–222. Document available online at http://www.cs.virginia.edu/ uigroup/publications/94/conferences/UIST/Survey.html.

Hinckley, K., Pausch, R., Goble, J. C., and Kassell, N. F. (1994b). Passive real-world interface props for neurosurgical visualization. In *Human Factors in Computing Systems, CHI '94 Conference Proceedings*, pages 452–458.

Hirose, M. and Yokoyama, K. (1992). VR application for transmission of synthetic sensation. In *Proceedings of ICAT '92*, pages 145–154.

Hirota, K. and Hirose, M. (1995). Simulation and presentation of curved surfaces in virtual reality environments through surface display. In *1995 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 211–216.

Hix, D. and Hartson, H. R. (1993). *Developing User Interfaces.* John Wiley & Sons, Inc.

Hix, D., Templeman, J., Amento, B., Schmidt-Nielsen, A., and Sibert, L. (1997). Empirical development and comparison of interaction techniques for panning and zooming in desktop virtual environments. Submitted to *Human Factors and Ergonomics Society Journal*.

Hix, D., Templeman, J., and Jacob, R. (1995). Pre-screen projection: From concept to testing of a new interaction technique. In *Human Factors in Computing Systems, CHI '95 Conference Proceedings*, pages 226–233.

Hodges, L. F., Rothbaum, B. O., Watson, B., Kessler, G. D., and Opdyke, D. (1996). A virtual airplane for fear of flying therapy. In *1996 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 86–93.

Hollander, A. J. and Furness, III, T. A. (1994). Perception of virtual auditory shapes. In *International Conference on Auditory Display (ICAD) '94 Proceedings*. Document available online at http://www.hitl.washington.edu/publications/p-94-3/hollander.html.

Hutchins, E. L. and Norman, D. A. (1986). *User-centered System Design*, pages 87–124. Lawrence Erlbaum Associates.

Ishizaki, S. (1996). Multiagent model of dynamic design. In *Human Factors in Computing Systems, CHI '96 Conference Proceedings*, pages 347–354.

Jacob, R., Sibert, L. E., McFarlane, D. C., and Mullen, Jr, M. P. (1994). Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction*, 1(1):3–26.

Jacobsen, L. (1996). Information from the *SIGGRAPH '96* Panel Session, "Cognition, Perception, and Experience in the Virtual Environment: Do You See What I See?".

Jacoby, R., Ferneau, M., and Humphries, J. (1994). Gestural interaction in a virtual environment. *SPIE*, 2177:355–364.

Kaczmarek, K. A. and Bach-Y-Rita, P. (1995). Tactile displays. In *Virtual Environments and Advanced Interface Design*, chapter 9, pages 349–414. Oxford University Press.

Kaiser Electro-Optics (1996). Information collected during a visit to Kaiser Electro-Optics, Carlsbad, CA, on July 10, 1996.

Karlgren, J., Bretan, I., Frost, N., and Jonsson, L. (1995). Interaction models, references, and interactivity in speech interfaces to virtual environments. In *Virtual Environments '95: Selected Papers of the Eurographics Workshops*, pages 149–159.

Kijima, R. and Hirose, M. (1996). Representative spherical plane method and composition of object manipulation methods. In *1996 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 195–202.

Kocain, D. F. and Task, H. L. (1995). Visually coupled systems hardware and the human interface. In *Virtual Environments and Advanced Interface Design*, chapter 6, pages 175–257. Oxford University Press.

Kontarinis, D. A. and Howe, R. D. (1995). Tactile display of vibratory information in teleoperation and virtual environments. *Presence: Teleoperators and Virtual Environments*, 4(4):387–402.

Krüger, W., Bohn, C.-A., Frölich, B., Schüth, H., Strauss, W., and Wesche, G. (1995). The responsive workbench: A virtual work environment. *IEEE Computer*, 28(7):42–48.

Liang, J. and Green, M. (1994). JDCAD: A highly interactive 3D modeling system. *Computers and Graphics*, 18(4):499–506.

Logitech (1997). Fremont, CA, USA (http://www.logitech.com/).

Lynch, K. (1960). *The Image of a City*. MIT Press.

MacKenzie, I. S. (1995). Input devices and interaction techniques for advanced computing. In *Virtual Environments and Advanced Interface Design*, chapter 11, pages 437–470. Oxford University Press.

Madritsch, F. and Gervautz, M. (1996). CCD-camera based optical beacon tracking for virtual and augmented reality. *Computer Graphics Forum*, 15(3):403–416.

Maggioni, C. (1993). A novel gestural input device for virtual reality. In *1993 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 118–124.

Malone, T. W. and Crowston, K. (1990). What is coordination theory and how can it help design cooperative work systems? In *Computer-Supported Cooperative Work (CSCW '90) Conference Proceedings*, pages 357–370.

Mapes, D. P. and Moshell, J. M. (1995). A two-handed interface for object manipulation in virtual environments. *Presence: Teleoperators and Virtual Environments*, 4(4):403–416.

McKenna, M. D. and Zeltzer, D. (1995). Three dimensional visual display systems for virtual environments. *Presence: Teleoperators and Virtual Environments*, 1(4):421–458.

McNeely, W. A. (1993). Robotic graphics: A new approach to force feedback for virtual reality. In *1993 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 336–341.

McNeely, W. A., Burdea, G. C., Hannaford, B., Hirose, M., Jacobsen, S., Salisbury, K., and Tachi, S. (1995). Panel: Whither force feedback. In *1995 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 226–227.

Mereu, S. W. (1995). Improving depth perception in 3D interfaces with sound. Department of Computer Science, Faculty of Math. MS Thesis. University of Waterloo. Waterloo, Ontario, Canada. Document available online at http://cgl.uwaterloo.ca/ swmereu/thesis.ps.

Mereu, S. W. and Kazman, R. (1996). Audio enhanced 3D interfaces for visually impaired users. In *Human Factors in Computing Systems, CHI '96 Conference Proceedings*, pages 72–78.

Meyer, K., Applewhite, H. L., and Biocca, F. A. (1992). A survey of position trackers. *Presence: Teleoperators and Virtual Environments*, 1(2):173–200.

Mine, M. R., Brooks, Jr., F. P., and Sequin, C. (1997). Moving objects in space: Exploiting proprioception in virtual-environment interaction. In *SIGGRAPH '97 Conference Proceedings*, pages 19–26.

MultiGen (1997). San Jose, CA, USA (http://www.multigen.com/.

National Center for Supercomputing Applications (1996). Information collected during a visit to NCSA, Urbana-Champaign, IL, in July, 1996.

Naval Postgraduate School (1996). Information collected during a visit to the Naval Postgraduate School, Monterey, CA, in July, 1996.

Naval Research Laboratory (1997). Information collected during a visit to NRL, Washington, D.C. in May, 1997.

Neale, D. C. and Carroll, J. M. (1997). The role of metaphors in user interface design. In Helander, M. G., Landauer, T. K., and Prabhu, P., editors, *Handbook of Human-Computer Interaction, 2nd edition*. Elsevier Science.

Nielsen, J. (1993). *Usability Engineering*. Academic Press.

Norman, D. A. (1990). *The Design of Everyday Things*. Doubleday: New York, New York.

Norman, D. A. and Draper, S. W. (1986). *Cognitive Engineering*, chapter 3. Lawerence Erlbaum Associates.

Oshhima, T., Yamamoto, H., and Tamura, H. (1996). Gaze-directed adaptive rendering for interacting with virtual space. In *1996 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 103–110.

Pausch, R., Burnette, T., Brockway, D., and Weiblen, M. E. (1996). Navigation and locomotion in virtual worlds via flight into hand-held miniatures. Document available online at http://www.cs.virginia.edu/ uigroup/publications/95/conferences/siggraph/paper.html.

Pausch, R., Shackelford, M. A., and Proffitt, D. (1993). A user study comparing head-mounted and stationary displays. In *1993 IEEE Symposium on Research Frontiers in Virtual Reality*, pages 41–45.

Polhemus Incorporated (1997). Colchester, VT, USA (http://www.polhemus.com/).

Poupyrev, I., Billinghurst, M., Weghorst, S., and Ichikawa, T. (1996). The go-go interaction technique: Non-linear mapping for direct manipulation in VR. In *Proceedings of ACM Symposium on User Interface Software & Technology (UIST '96)*, pages 79–80. Document available online at http://www.hitl.washington.edu/publications/.

Rekimoto, J. (1995). A vison-based head tracker for fish tank virtual reality - VE without the head gear. In *1995 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 94–100.

Richard, P., Birebent, G., Coiffet, P., Burdea, G., Gomex, D., and Langrana, N. (1996). Effect of frame rate and force feedback on virtual object manipulation. *Presence: Teleoperators and Virtual Environments*, 5(1):95–108.

Sarcos Research Corportation (1997). Salt Lake City, UT, USA (http://www.sarcos.com/).

SensAble Technologies, Inc. (1997). Cambridge, MA, USA (http://www.sensable.com/).

Shimoga, K. B. (1993). A survey of perceptual feedback issues in dexterous telemanipulation: Part I. finger force feedback. In *1993 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 263–270.

Shneiderman, B. (1992). *Designing the User Interface: Strategies for Effective Human Computer Interaction*. Addison-Wesley.

Singh, G. and Feiner, S. K. (1995). *Transactions on Computer-Human Interaction — Special Issue on Virtual Reality Software and Technology*, volume 2, pages 177–178. ACM.

Slater, M., Steed, A., and Usoh, M. (1995a). The virtual treadmill: A naturalistic metaphor for navigation in immersive virtual environments. In *Virtual Environments '95: Selected Papers of the Eurographics Workshops*, pages 135–148. Document available online at http://www.dcs.qmw.ac.uk/.

Slater, M., Usoh, M., and Steed, A. (1995b). Taking steps: The influence of a walking technique on presence in virtual reality. *ACM Transactions on Computer-Human Interaction*, 2(3):201–219.

Snow, M. (1996). Charting presence in virtual environments and its effects on performance. Department of Industrial & Systems Engineering. Ph.D. Dissertation. Virginia Tech. Blacksburg, VA.

Sowizal, H. A. (1994). Interacting with virtual environments using augmented virtual tools. *SPIE*, 2177:409–416.

Sowizral, H. A. and Barnes, J. C. (1993). Tracking position and orientation in a large volume. In *1993 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 132–140.

Spacetec IMC (1997). Lowell, MA, USA (http://www.spacetec.com/).

SRI (1996). Information collected during a visit to SRI International, Menlo Park, CA, on July 15, 1996.

Stanney, K. (1995). Realizing the full potential of virtual reality: Human factors issues that could stand in the way. In *1995 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 28–34.

State, A., Livingston, M. A., Garrett, W. F., Hirota, G., Whitton, M. C., Pisano, E. D., and Fuchs, H. (1996). Technologies for augmented reality systems: Realizing ultrasound-guided needle biopsies. In *SIGGRAPH '96 Conference Proceedings*, pages 439–446.

Stoakley, R., Conway, M. J., and Pausch, R. (1995). Virtual reality on a WIM: Interactive worlds in miniature. In *Human Factors in Computing Systems, CHI '95 Conference Proceedings*. Document available online at http://www.cs.cmu.edu/ stage3/publications/95/conferences/chi/paper.html.

Strickland, D., Patel, A., Stovall, C., Palmer, J., and McAllister, D. (1994). Self tracking of human motion for virtual reality systems. *SPIE*, 2177:278–287.

Sturman, D. J. (1992). Whole-hand input. Ph.D. Thesis, Media Arts & Science, MIT.

Sturman, D. J. and Zeltzer, D. (1994). A survey of glove-based input. *Computer Graphics & Applications*, 14(1):30–39.

Sturman, D. J., Zeltzer, D., and Pieper, S. (1989). Hands-on interaction with virtual environments. In *Proceedings of ACM Symposium on User Interface Software & Technology (UIST '89)*, pages 19–24.

Su, S. A. and Furuta, R. (1994). A logical hand device in virtual environments. In *VRST '94: Virtual Reality Software and Technology*, pages 33–44.

Sutherland, I. (1968). A head-mounted three dimensional display. In *Proceedings of the Fall Joint Computing Conference*, pages 757–764.

Templeman, J. (1996). Enhancing user control of VE walkthroughs. Presentation to Office of Naval Research.

Templeman, J. (1997a). Personal Communication.

Templeman, J. (1997b). Manipulation of object sets in synthetic environments. Unpublished Naval Research Laboratory document.

to the Interface, A. I. (1996). M. billinghurst and j. savage. In *1996 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 168–175.

Trias, T. S., Chopra, S., Reich, B. D., Moore, M. B., Badler, N. I., Webber, B. L., and Geib, C. W. (1996). Decision networks for integrating the behaviors of virtual agents and avatars. In *1996 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 156–162.

UNC (1996). Information from the University of North Carolina - Chapel Hill Exhibit in the *SIGGRAPH '96* Digital Bayou.

University of Illinois at Chicago Electronic Visualization Laboratory (1997). Chicago, IL, USA (http://evlweb.eecs.uic.edu/EVL/).

University of Washington (1996). Information collected during a visit to the University of Washington's HITLab, Seattle, WA, on July 24, 1996.

UVA (1996). Information gathered during a visit to the University of Virginia, Charlottesville, VA, in June, 1996.

Verlinden, J., Bolter, J., and Van der Mass, C. (1993). Voice annotation: Adding verbal information to virtual environments. In *Proceedings of the European Simulation Symposium*, pages 60–69.

Virtual Space Devices, Inc. (1997). Bloomington, MN, USA (http://www.vsdevices.com/).

Virtual Technologies, Inc. (1997). Palo Alto, California, USA (http://www.virtex.com/).

Vivid (1997). Big head racer. As seen in *SIGGRAPH '97* Electric Garden. See SIGGRAPH'97 Visual Proceedings or http://www.vividgroup.com.

Waldrop, M. S., Pratt, S. M., Pratt, D. R., McGhee, R. B., Falby, J. S., and Zyda, M. J. (1995). Real-time upper body articulation of humans in a networked interactive virtual environment. Document available online at http://www-npsnet.cs.nps.navy.mil/npsnet/publications.html.

Ware, C., Arthur, K., and Booth, K. S. (1993). Fish tank virtual reality. In *Proceedings of INTERCHI'93 Conference on Human Factors in Computing Systems*, pages 37–41.

Ware, C. and Balakrishnan, R. (1994). Reaching for objects in VR displays: Lag and frame rate. *ACM Transactions on Computer-Human Interaction*, 1(4):331–356.

Ware, C. and Osborne, S. (1990). Exploration and virtual camera control in virtual three dimensional environments. *Computer Graphics*, 24(2):175–183.

Waters, R. C., Anderson, D. B., Barrus, J. W., Brogan, D. C., Casey, M. A., McKeown, S. G., Nitta, T., Sterns, I. B., and Yerazunis, W. S. (1997). Diamond park and spline: A social virtual reality system with 3D animation, spoken interaction, and runtime modifiability. Technical Report TR96-02a, MERL — A Mitsubishi Electronic Research Laboratory. Document available online at http://www.merl.com/reports/TR96-02a/index.html.

Wexelblat, A. (1995). An approach to natural gesture in virtual environments. *ACM Transactions on Computer-Human Interaction*, 2(3):179–200.

Wickens, C. D. and Baker, P. (1995). Cognitive issues in virtual reality. In *Virtual Environments and Advanced Interface Design*, chapter 13, pages 516–541. Oxford University Press.

Wiker, S. F., Vandereheiden, G., Lee, S., and Arndt, S. (1991). Development of tactile mice for blind access to computers: Importance of stimulation locus, object size, and vibrotactile display resolution. In *Proceedings Human Factors Society 35th Annual Meeting*, pages 708–712.

Yoshida, A., Rolland, J. P., and Reif, J. H. (1995). Design and applications of a high-resolution insert head-mounted display. In *1995 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 84–93.

Zhai, S. (1995). Human performance in six degree of freedom input control. Ph.D. Thesis, University of Toronto. Document available online at http://vered.rose.toronto.edu/people/shumin.html.

Zhai, S. (1997). The EGG (Elastic General-purpose Grip). Document available online at http://vered.rose.toronto.edu/people/shumin_dir/EGG.html.

Zhai, S., Buxton, W., and Milgram, P. (1994). The “silk cursor”: Investigating transparency for 3d target acquisition. In *Human Factors in Computing Systems, CHI ’94 Conference Proceedings*, pages 459–464. Document available online at http://vered.rose.toronto.edu/people/shumin_dir/papers/CHI94.html.

Zhai, S. and Milgram, P. (1993a). Human performance evaluation of isometric and elastic rate controllers in a 6 dof tracking task. *SPIE*, 2057:130–141. Document available online at http://vered.rose.toronto.edu/people/shumin_dir/papers/SPIE93.html.

Zhai, S. and Milgram, P. (1993b). Human performance evaluation of manipulation schemes in virtual environments. In *1993 IEEE Virtual Reality Annual International Symposium Proceedings*, pages 155–161. Document available online at http://vered.rose.toronto.edu/people/shumin_dir/papers/VRAIS93.html.

Zhai, S. and Milgram, P. (1994). Input techniques for hci in 3D environments. In *Human Factors in Computing Systems, CHI ’94 Conference Proceedings Companion*, pages 85–86. Document available online at http://vered.rose.toronto.edu/people/shumin_dir/papers/CHI94Poster.html.

Zhai, S., Milgram, P., and Buxton, W. (1996). The influence of muscle groups on performance of multiple degree-of-freedom input. In *Human Factors in Computing Systems, CHI '96 Conference Proceedings*, pages 308–315. Document available online at http://vered.rose.toronto.edu/people/shumin_dir/papers/CHI96/sz_txt.html.

Zhai, S. and Senders, J. W. (1997). Investigating coordination in multidegree of freedom. In *Proceedings of the Human Factors and Ergonomics Society 41st Annual Meeting*, pages 1249–1262.

Zyda, M. J., Pratt, D. R., Shirley, M., Barham, P. T., and Falby, J. S. (1995). NPSNET-HUMAN: Inserting the human into the networked synthetic environment. In *Proceedings of the 13th DIS Workshop*, page unpaginated.

Appendix A VE Labs Visited During Investigative Visits

- University of Virginia, Charlottesville VA
- Naval Research Lab, Washington DC
- High-Techsplanations Inc., Rockville MD
- George Washington University, Washington DC
- National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign IL
- EDS Virtual Reality Center, Detroit MI
- NRAD, San Diego CA
- Kaiser Electro-Optics, Carlsbad CA
- Naval Postgraduate School, Monterey CA
- Fakespace, Menlo Park CA
- Lockheed Martin, Palo Alto CA
- SRI International, Menlo Park CA
- University of Washington, Seattle WA
- Boeing, Seattle WA

Appendix B Editor's Comments and Suggestions

The following is a list of editor's comments and suggestions. Note that each comments is labeled as either "current" or "future", distinguishing those comments which were integrated into the present version of the taxonomy from those comments which we hope to integrate in future revisions.

The following comments address the taxonomy's structure:

Editor's Comment/Suggestion	***Integration***
Provide a labeling scheme for the specific usability suggestions	present
Facilitate access from context driven discussion back into tables	present
Structure specific usability suggestions as ordered lists as opposed to tables	future
Support dynamic document structuring	future
Provide a notion of (specific usability suggestion) priorities that might come into play.	future
Provide well-defined rules as to how to develop classifications from the taxonomy, so that I could take my VE system and classify it according to the categories you set out.	future
Order four main taxonomy areas differently.	future
Provide access to taxonomy content from any point within Norman's theory of action.	future
Develop taxonomy as simply the overview diagram and specific usability suggestions. Demote context driven discussion to taxonomy companion.	future

The following comments address the taxonomy's content:

Editor's Comment/Suggestion	***Integration***
Fix general grammar and spelling errors	present
Include editor's specific usability suggestions and references	present
Clearly state that the taxonomy is are mostly talking about usability for human performance in VEs in general (distinguished from any notion of training transfer).	present
State that the taxonomy is not complete (nor will it ever be; taxonomies can't ever prove completeness), and that it's early enough in the development and investigation process that parts of it may be incorrect.	present
Point out the few most well-known instances where the literature contradicts itself.	present
Further develop the section on navigation.	present
Discuss when not to use stereopsis (as opposed to simple when to use stereopsis).	future
Include a more detailed discussion on virtual menus.	future

Appendix C Acronyms

BOOM	Binocular Omni-Oriented Monitor
CAD	Computer-Aided Design
CAVE™	Cave Automatic Virtual Environment
CHI	Computer-Human Interaction
CSCW	Computer-Supported Cooperative Work
DIVE	Distributed Interactive Virtual Environments
DOF	Degrees of Freedom
EVL	Electronic Visualization Laboratory (University of Illinois at Chicago)
GM	General Motors
HCI	Human-Computer Interaction
HT	High Techsplanations Inc.
IPORT	Individual Portal
MIT	Massachusetts Institute of Technology
NCSA	National Center for Supercomputing Applications
NPS	Naval Postgraduate School
NRL	Naval Research Lab
ODT	Omni-Directional Treadmill
SICS	Swedish Institute of Computer Science

SID Spatially Immersive Display

SIGGRAPH ACM Special Interest Group on Graphics

UNC-CH University of North Carolina at Chapel Hill

VE Virtual Environment

VMD Virtual Model Display

VRAIS Virtual Reality Annual International Symposium

WIM Worlds in Miniature

WWW World Wide Web

VITA

Joseph L. Gabbard

Joseph L. Gabbard, son of Joseph L. Gabbard and Judith Ann Rainer, was born in Fairfax, Virginia on the 11th day of March, 1969. Raised mostly in Loudoun County, Virginia, he lived for approximately five years in Little Rock, Arkansas, where he was fortunate enough to have had several soul building experiences camping on Lake Ouachita nestled in the beautiful Ouachita National Forest.

He holds an M.S. and B.S. in Computer Science and a B.A. in Sociology both from Virginia Polytechnic Institute and State University (Virginia Tech) in Blacksburg, Virginia.

Joseph's research interests are strongly focused on the nature of human-computer interaction within virtual environments (VEs). In particular, he has developed a taxonomy of usability characteristics in VEs, identifying usability issues critical to human performance within VEs. Currently, he is researching how usability engineering and evaluation methods can be infused into VE design, evaluation, and training.

Joseph L. Gabbard currently resides in Blacksburg, Virginia and can be reached through http://csgrad.cs.vt.edu/ jgabbard/.